



HAL
open science

Pointwise approach for texture analysis and characterization from very high resolution remote sensing images

Minh-Tan Pham

► **To cite this version:**

Minh-Tan Pham. Pointwise approach for texture analysis and characterization from very high resolution remote sensing images. Computer Vision and Pattern Recognition [cs.CV]. TELECOM Bretagne, 2016. English. NNT: . tel-01464333v1

HAL Id: tel-01464333

<https://theses.hal.science/tel-01464333v1>

Submitted on 10 Feb 2017 (v1), last revised 26 Sep 2017 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 2016telb0403



Sous le sceau de l'Université Bretagne Loire

Télécom Bretagne

En accréditation conjointe avec l'École Doctorale – SICMA

Thèse de Doctorat

Mention : *STIC – Sciences et Technologies de l'Information et des Communications*

Pointwise approach for texture analysis and characterization from very high resolution remote sensing images

présentée par

Minh-Tan PHAM

Unité de recherche : **Télécom Bretagne – Image et Traitement de l'Information**
Laboratoire : **Lab-STICC, Pôle CID, Équipe TOMS**
Entreprise partenaire : **Centre National d'Etudes Spatiales (CNES)**
Directeur de thèse : **Grégoire Mercier**
Encadrant de thèse : **Julien Michel**

Soutenue le **20 septembre 2016** devant le jury composé de:

M. Emmanuel Trouvé - Professeur, Université Savoie Mont Blanc	Président
M. Yannick Berthoumieu - Professeur, IPB/ENSEIRB-Matmeca	Rapporteur
M. Philippe Carré - Professeur, Université de Poitiers	Rapporteur
Mme. Florence Tupin - Professeure, Télécom ParisTech	Examineur
M. Grégoire Mercier - Professeur, Télécom Bretagne	Examineur
M. Julien Michel - Ingénieur, CNES	Examineur

Acknowledgements

During my 3-year PhD work, I have had the opportunity to learn a great amount of knowledge and experience in the fields of computer vision, image processing as well as remote sensing. I firstly would like to thank the French Space Agency (CNES) and the Brittany Region for funding my work; thank Télécom Bretagne, Laboratory LabSTICC and Télécom Paristech for their supports and materials. More importantly, I have got a lot of chances to meet and work with many persons to whom I would like to express my sincere gratitude.

I am so grateful to Prof. Yannick Berthoumieu and Prof. Philippe Carré for reviewing my thesis. I appreciate their interest in my work as well as all of their insightful comments and suggestions.

I would like to thank Prof. Florence Tupin and Prof. Emmanuel Trouvé for taking part in my PhD defense at Télécom ParisTech. I am thankful for their careful reading on my manuscript as well as their positive comments during my defense.

I thank Julien Michel for his remote supervision and advice. Thanks for guiding me using OTB during my 2-week stay in CNES at the beginning of my second year. It was a pleasure to work with him and the team in CNES.

I am especially grateful to Prof. Grégoire Mercier for his supervision which is so effective and extremely respectable. I will never forget all the time we spent together on discussing new ideas, working on and revising papers, planning for and going to conferences, etc. His guidance makes me feel so comfortable, motivated and confident! I always want to say to him: "Merci beaucoup Greg! Tu es mon super super chef! Banana!"

A special thank you goes to Lionel Bombrun (IMS Bordeaux) and Olivier Regniers (I-SEA) for their collaborations during the final stage of my work. Also, I am grateful to Josiane Zerubia (INRIA Sophia-Antipolis), Philippe Durand (CNES), Andreas Reighber (DLR, Germany), Marco Lavallo (NASA, US), Thomas Ainsworth (NRL, US) for their fruitful discussions and for all image data they provided.

I also would like to thank all my professors from Télécom Bretagne, and so far from Ho Chi Minh University of Technology as well as Tien Giang high school for gifted students, for all knowledge and skills they provided to me. I am thankful to Prof. Sylvie Daniel from Université Laval, Québec city for guiding me during my master internship.

Many sincere thanks go to my dear friends and colleagues from Brest, from other parts of France, from Canada, from US and from Vietnam! A very special thank you is given to my Vietnamese friends in "4bis rue de Kérangoff", "Bờ Zét Hội" and "Hội cầu lông Brest" for all the great and joyful moments we spent together which will be unforgettable in my life.

I finally leave this longest paragraph to thank my family and my love! I am deeply grateful to them for all their support and faith, not only within my PhD, but also in every step during my whole life. They always make me strong and motivated. Thank "Cha", "Mẹ", "Anh 2", "Phước", "Trần", "Tú" for always being with me. Thank "Gia Đình cô Phi chú Nghiệp" (my second family) for considering me as a family member. And finally, thank "Hương Trà", my LOVE, for always waiting, encouraging and trusting me ... *"OXYBXNtrĐ!"*

Contents

Acknowledgments	i
Abstract	vii
Acronyms	ix
Résumé étendu	1
Introduction	1
Contexte de l'étude	1
Motivation et objectif de la thèse	2
Principe du traitement d'image par points clés	3
Traitement du signal sur graphe pour la caractérisation des textures	4
Analyse spectrale par graphe pour la segmentation en texture	6
Analyse ponctuelle de texture pour l'analyse des changements et des déplacements	8
Approche ponctuelle et texture structurée	13
Covariance locale sur points clés	13
Descripteur local sur points clés	16
Conclusion et perspectives	20
Perspectives	20
I Introduction	21
1 General introduction	23
1.1 Context of the thesis	23
1.2 Classical approaches for texture analysis	24
1.3 Motivation and objective	26
1.4 Thesis overview and organization	27
2 Introduction to a pointwise approach for texture analysis	29
2.1 Introduction	29
2.2 Extraction of characteristic points	29
2.3 Sensitivity to parameter	33
2.4 Why not other types of keypoints?	33
2.5 Conclusion	35

II	Pointwise approach combined with graph theory	37
3	Review of signal processing on graphs	39
3.1	Introduction	39
3.2	Weighted graph and characteristics	40
3.2.1	Definition	40
3.2.2	Spectral domain of graph	41
3.2.3	Spectral graph clustering	41
3.3	Signal processing on graphs	42
3.3.1	On vertex domain	42
3.3.2	On spectral domain	43
3.4	Illustration: graph for image	44
3.4.1	Local graph for image decomposition	44
3.4.2	Non-local graph for image denoising	46
3.5	Motivation: graph for keypoints	46
3.6	Conclusion	48
4	Pointwise graph-based texture description for VHR optical image classification	49
4.1	Introduction	50
4.2	Texture description using a pointwise approach	51
4.3	Proposed classification framework	53
4.3.1	Extraction of characteristic points for image representation and description	55
4.3.2	Weighted graph construction	56
4.3.3	Textural feature extraction	57
4.3.4	Clustering algorithm for unsupervised classification	59
4.4	Experimental results	59
4.4.1	Datasets	59
4.4.2	Experimental setup	59
4.4.3	Classification results	61
4.4.4	Performance in time consumption	65
4.4.5	Sensitivity of the proposed method	66
4.5	Conclusion	67
5	Pointwise graph-based texture tracking for change detection from SAR images	69
5.1	Introduction	70
5.1.1	Context: change detection using SAR images	70
5.1.2	Motivation of the chapter	71
5.2	Texture tracking for change detection based on a pointwise graph-based approach	71
5.2.1	Proposed framework	71
5.2.2	Extraction of local extrema points adapted to SAR images	72
5.2.3	Weighted graph construction from keypoints	74
5.2.4	Change measure generation	75
5.3	Experiments	76
5.3.1	Data description and evaluation criteria	76
5.3.2	Change detection results	77
5.3.3	Sensitivity to parameters	81
5.3.4	Comparison with the NLM model	86
5.3.5	Analysis of the algorithm complexity	87

5.3.6	Exploitation of the local minimum pixels	89
5.4	Application to glacier displacement detection	90
5.4.1	Proposed method	90
5.4.2	Preliminary results	91
5.5	Conclusion	92
III	Pointwise approach combined with structural features	95
6	PW-COG: A pointwise covariance descriptor for texture characterization	97
6.1	Introduction	98
6.1.1	Motivation of the chapter	98
6.1.2	Chapter overview	99
6.2	Pointwise construction for covariance-based descriptors	99
6.2.1	Region covariance matrix of oriented gradients	99
6.2.2	Pointwise construction of COG	100
6.2.3	Adapted distance metric for covariance descriptors	101
6.3	Application to texture-based VHR image classification	102
6.4	Experimental study	103
6.4.1	Data description	103
6.4.2	Experiment setup	104
6.4.3	Results and discussion	107
6.4.4	Sensitivity to parameters	116
6.4.5	Sensitivity to the distance metric	117
6.5	Conclusion	118
7	LED: An efficient local extrema descriptor for texture retrieval from VHR images applied to vineyard detection	119
7.1	Introduction	120
7.1.1	Motivation of the chapter	120
7.1.2	Chapter organization	120
7.2	Texture retrieval from VHR optical images using the local extrema descriptor	121
7.2.1	Extraction of LED	121
7.2.2	Dissimilarity measure for retrieval	124
7.2.3	Retrieval algorithm	125
7.3	Retrieval experiments	127
7.3.1	Studied sites and data	127
7.3.2	Experiment setup	128
7.3.3	Results and discussion	129
7.3.4	Sensitivity to parameters	130
7.4	Application to vineyard detection based on supervised classification	133
7.4.1	Supervised classification algorithm	133
7.4.2	Experimental results	134
7.5	Conclusion	140

IV Conclusion	141
8 Conclusions and Perspectives	143
8.1 Conclusion	143
8.2 Perspectives	144
List of Figures	148
List of Tables	150
Bibliography	157
A Evaluation criteria for detection tasks	159
B Author Publications	161

Abstract

Abstract: This thesis work proposes a novel pointwise approach for texture analysis in the scope of very high resolution (VHR) remote sensing imagery. This approach takes into consideration only characteristic pixels, not all pixels of the image, to represent and characterize textural features. Due to the fact that increasing the spatial resolution of satellite sensors leads to the lack of stationarity hypothesis in the acquired images, such an approach becomes relevant since only the interaction and characteristics of keypoints are exploited. Moreover, as this technique does not need to consider all pixels inside the image like classical dense approaches, it is more capable to deal with large-size image data offered by VHR remote sensing acquisition systems. In this work, our pointwise strategy is performed by exploiting the local maximum and local minimum pixels (in terms of intensity) extracted from the image. It is integrated into several texture analysis frameworks with the help of different techniques and methods such as the graph theory, the covariance-based approach, the geometric distance measurement, etc. As a result, a variety of texture-based applications using remote sensing data (both VHR optical and radar images) are tackled such as image retrieval, segmentation, classification, and change detection, etc. By performing dedicated experiments to each thematic application, the effectiveness and relevance of the proposed approach are confirmed and validated.

Keywords: *remote sensing imagery, texture analysis, pointwise approach, graph theory, texture retrieval, image classification, change detection.*

Résumé : Ce travail de thèse propose une nouvelle approche ponctuelle pour l'analyse de texture dans l'imagerie de télédétection à très haute résolution (THR). Cette approche ne prend en compte que des points caractéristiques, et non pas tous les pixels dans l'image, pour représenter et caractériser la texture. Avec l'augmentation de la résolution spatiale des capteurs satellitaires, les images THR ne vérifient que faiblement l'hypothèse de stationnarité. Une telle approche devient donc pertinente étant donné que seuls l'interaction et les caractéristiques des points-clés sont exploitées. De plus, puisque notre approche ne considère pas tous les pixels dans l'image comme le font la plupart des méthodes denses de la littérature, elle est plus à-même de traiter des images de grande taille acquises par des capteurs THR. Dans ce travail, la méthode ponctuelle est appliquée en utilisant des pixels de maxima locaux et minima locaux (en intensité) extraits à partir de l'image. Elle est intégrée dans plusieurs chaînes de traitement en se fondant sur différentes techniques existantes telles la théorie des graphes, la notion de covariance, la mesure de distance géométrique, etc. En conséquence, de nombreuses applications basées sur la texture sont abordées en utilisant des données de télédétection (images optiques et radar), telles l'indexation d'images, la segmentation, la classification et la détection de changement, etc. En effectuant des expériences dédiées à chaque application thématique, la pertinence et l'efficacité du cadre méthodologique proposé sont confirmées et validées.

Mots clés : *imagerie de télédétection, analyse de texture, approche ponctuelle, théorie des graphes, indexation de texture, classification, détection de changement.*

Acronyms

ACA	Average Classification Accuracy
AR	Auto-regressive
ARR	Average Retrieval Rate
COG	Covariance matrix of Oriented Gradients
CM	Change Measure
CKLD	Cumulant-based Kullback Leibler Divergence
DCOG	Dense Covariance matrix of Oriented Gradients
DFT	Discrete Fourier Transform
DI	Different Image
EMD	Empirical Mode Decomposition
FA	False Alarm
GCM	Gaussian Copula-based Model
GD	Good Detection
GFB	Gabor Filter Bank
GLCM	Gray Level Cooccurrence Matrix
GLOH	Gradient Location and Oriented Histogram
HOG	Histogram of Oriented Gradients
HSOG	Histogram of Second Order Gradients
IB	Intensity Based
KI	Kittler Illingworth
KLD	Kullback Leibler Divergence
kNN	k-Nearest Neighbors
LBP	Local Binary Pattern
LED	Local Extrema Descriptor

LH	Local Histogram
LN	Local Neighborhood
LR	Log-Ratio
LRD	Log-Ratio Detector
MD	Missed Detection
MGM	Multivariate Gaussian Model
MRD	Mean-Ratio Detector
MRF	Markov Random Field
NLM	Non-Local Mean
OA	Overall Accuracy
OCA	Overall Classification Accuracy
PCA	Principal Component Analysis
PDF	Probability Density Function
PJD	Projection-based Jeffrey Detector
PKLD	Pearson-based Kullback Leibler Divergence
PW	Pointwise
PW-COG	Pointwise Covariance matrix of Oriented Gradients
ROC	Receiver Operating Characteristics
ROI	Region Of Interest
SAR	Synthetic Aperture Radar
SCM	Sample Covariance Matrix
SGC	Spectral Graph Clustering
SGFT	Spectral Graph Fourier Transform
SGWT	Spectral Graph Wavelet Transform
SIFT	Scale Invariant Feature Transform
SIRV	Spherically Invariant Random Vectors
SURF	Speeded Up Robust Features
TCT	Total Computation Time
TE	Total Error
VHR	Very High Resolution
WLD	Weber Local Descriptor

Résumé étendu

Introduction	1
Contexte de l'étude	1
Motivation et objectif de la thèse	2
Principe du traitement d'image par points clés	3
Traitement du signal sur graphe pour la caractérisation des textures	4
Analyse spectrale par graphe pour la segmentation en texture	6
Analyse ponctuelle de texture pour l'analyse des changements et des déplacements	8
Approche ponctuelle et texture structurée	13
Covariance locale sur points clés	13
Descripteur local sur points clés	16
Conclusion et perspectives	20
Perspectives	20

Note : Theses completed in a French institution but written in another language are required to provide a French abstract.

Note : Les thèses accomplies dans un établissement français mais écrits dans une autre langue doivent fournir un résumé français.

Introduction

Contexte de l'étude

Nous nous sommes intéressés dans ce travail de thèse à l'analyse et à la caractérisation des textures locales dans les images de télédétection optiques et radar à Très Haute Résolution spatiale (THR). Ces images THR engendrent en effet des textures qui ne répondent plus aux hypothèses de stationnarité classiques et valables avec des images de résolution décimétrique ; les textures engendrées sont également potentiellement de taille beaucoup trop petite pour que les méthodes classiques puissent y extraire des paramètres de caractérisation fiable.

Dans la littérature, nous pouvons regrouper les méthodes d'analyse de texture selon quatre grandes catégories (cette classification est notamment extraire des articles de synthèses suivants : [1–3]), même si un algorithme d'analyse complète a recours à des techniques qui proviennent de plusieurs de ces catégories.

Approches statistiques : On a recours dans ces approches à une modélisation des niveaux de gris de l'image, plus exactement de son histogramme, par des lois de distribution statistique [4]. Si l'approche n'est plus du premier ordre mais d'un ordre plus élevé, on obtient une approche fondée sur l'autocorrélation du signal [5], ainsi que la méthode des matrices de co-occurrences [6], qui nous servira de référence principale.

Approches fondées sur un modèle : On considère ici que les textures sont générées par un modèle, qu'il soit d'inspiration fractale [7–9] ou stochastique (auto-régressif [10], markovien [11], ...). Bien souvent, dans ces approches, on considère que les valeurs de pixels dépendent principalement du voisinage immédiat et non pas les pixels de l'image entière.

Approches par transformation : Les approches par transformations permettent de séparer les caractéristiques des textures selon des échelles particulières. Ces échelles sont directement associées à des notions de fréquences dans le cas de la transformée de Fourier ou en Cosinus Discrets. Elles sont plus liées à des notions de filtrages passe-bande dans le cas de la transformée de Gabor [12, 13] ou des approches fondées sur la transformée en ondelettes [14–17].

Approches structurelles : Ces approches, beaucoup moins répandues pour la caractérisation, sont fondées sur la description d'un arrangement spatial de textures primaires. Ces dictionnaires de base sont fondés sur des motifs binaires [18], des approches morphologiques [19] ou des notions plus explicites de texton [20].

Motivation et objectif de la thèse

Lors de ce travail de thèse, nous avons tenté de définir une nouvelle approche d'analyse de texture qui soit capable, d'une part, de contourner la restriction d'une stationnarité, même locale, des textures, d'autre part, de traiter de très gros volumes de données, les images de télédétection actuellement fournies par les capteurs THR étant bien souvent de taille supérieure à 10000×10000 pixels.

Voici donc les principales contributions de cette thèse :

- La texture est à présent appréhendée à travers une notion de points caractéristiques (*Pointwise approach*), ou points clés, qui la distingue des approches qualifiées de denses lorsqu'elles ont recourt à des fenêtres glissantes.
- Nous définissons alors des outils permettant de traiter ces ensembles de points clés dans une image pour en extraire des caractéristiques de texture.
- L'analyse de texture est alors déclinée selon différents contextes
 - Segmentation et classification de textures pour les images optiques panchromatiques et multi-spectales dans un contexte agricole, forestier, urbain et péri-urbain.
 - Suivi de texture appliqué à la détection de changement ainsi qu'aux mesures de déplacement (de glacier).
 - Reconnaissance d'image à base de texture, ce qui correspond à une version supervisée du premier point.
- Ce travail est associé à une production scientifique conséquente avec 5 publications en journaux de rang A et une soumission en cours de relecture.

Principe du traitement d'image par points clés

Les approches d'échantillonnage compressé (Compressive Sensing [21]), de décomposition modale empirique (EMD [22–24]), et dans une moindre mesure les algorithmes fondés sur les SIFT [25] ou SURF [26, 27], nous ont montré qu'il était possible de traiter des données sans suivre rigoureusement les principes de l'échantillonnage de Shannon [28]. Des points clés, correspondant à un sous-échantillonnage non-régulier du signal initial selon un critère donné, permettrait de résoudre des problèmes de caractérisation, segmentation et d'analyse de texture et recalage avec des performances équivalentes ou meilleures que les approches *denses* de la littérature.

Le parti pris développé ici repose sur le fait de pouvoir caractériser une texture à partir de points clés extraits de l'image. Ces points clés doivent cependant être représentatifs de situations locales de l'image (en terme par exemple de contraste, de finesse éventuelle des détails) et couvrir de façon homogène la surface imagée de façon à ne pas avoir de trou dans la représentation des données (comme pourrait le faire un détecteur SIFT par exemple). Aussi les critères des maxima et minima locaux ont été retenus pour une telle caractérisation. La figure 1 illustre d'ailleurs la pertinence de l'approche "max/min locaux" dans la caractérisation des textures.

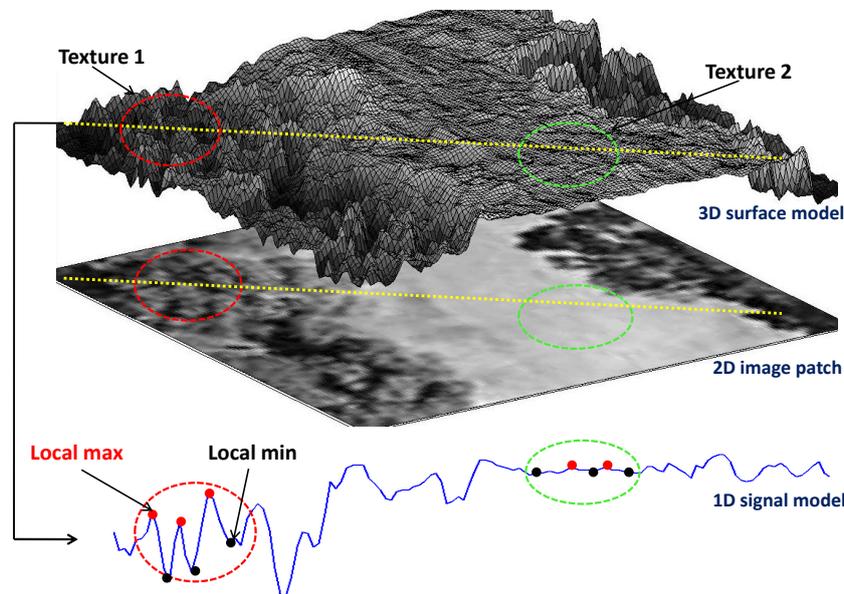


Figure 1 – Représentation des textures d'une image Pléiades à travers un modèle 3D où le niveau de gris est assimilé à une hauteur. Le transect illustre la pertinence de considérer des maxima et minima locaux dans la représentation locale d'une texture.

Plus formellement, on va donc définir deux ensembles de points, extraits d'une image I (considérée comme monochromatique) : $\mathcal{S}_\omega^{\max}(I)$ et $\mathcal{S}_\omega^{\min}(I)$ qui définissent respectivement les maxima et minima locaux (à travers une fenêtre glissante de taille $\omega \times \omega$) de l'image I .

$$\mathcal{S}_\omega^{\max}(I) = \left\{ p/I(p) = \max_{q \in \mathcal{N}_\omega(p)} I(q) \right\}, \quad (1)$$

$$\mathcal{S}_\omega^{\min}(I) = \left\{ p/I(p) = \min_{q \in \mathcal{N}_\omega(p)} I(q) \right\}, \quad (2)$$

où $I(p)$ représente la valeur du pixel p de l'image I et où $\mathcal{N}_\omega(p)$ caractérise le voisinage de taille $\omega \times \omega$ centré sur le point p . La figure 2 représente la distribution de tels points clés sur une image Pleiade ayant des textures orientées ou quelconques.



Figure 2 – Représentation de variété des points clés de $\mathcal{S}_\omega^{\max}(I)$ (en rouge) et $\mathcal{S}_\omega^{\min}(I)$ (en bleu) sur une image Pléiades ayant des textures locales différentes.

L’approche fondée sur les notions d’extrema locaux peut paraître extrêmement simpliste. Elle a cependant l’avantage de fournir des points d’intérêt de façon homogène quelque soit le type de texture. La figure 3 montre que l’approche par extrema garanti une représentation homogène des textures quelque soit le type de texture sous-jacent...

Traitement du signal sur graphe pour la caractérisation des textures

Avec la définition des points d’intérêt vue, de façon générale, à la section précédente, la définition du graphe qui regroupe ces points d’intérêt est assez naturelle et est bien résumée à la figure 4. Pour définir un graphe, il y a trois étapes à suivre :

- 1) Définition des nœuds. Comme l’illustre la figure 4, il suffit de s’appuyer sur les points d’intérêt... Dans certaines applications, seuls les maxima locaux peuvent être considérées : $\mathcal{S}_\omega^{\max}(I)$.
- 2) La définition des arêtes est liée à la taille d’une fenêtre d’analyse de taille $W \times W$. Deux pixels p et q sont connectés dans un graphe, si $q \in \mathcal{N}_W(p)$ (le point de vue étant symétrique, on a également dans ce cas $p \in \mathcal{N}_W(q)$ ce qui implique que le graphe est symétrique dans sa structure). Si $W > \omega$ on a toujours au moins une arête par nœud.
- 3) Le poids des arêtes est fixé de façon relativement classique en considérant une distance entre les pixels d’intérêt :

$$w(p, q) = e^{-\gamma \text{dist}(p, q)}. \quad (3)$$

$\text{dist}(p, q)$ est la distance qui sépare les points clés p et q de l’image. Le jeu consiste ici à définir une distance idoine de façon à associer correctement les points clés pour une caractérisation efficace de la texture. La définition idoine de la fonction de distance sera notamment le cœur des sections suivantes.

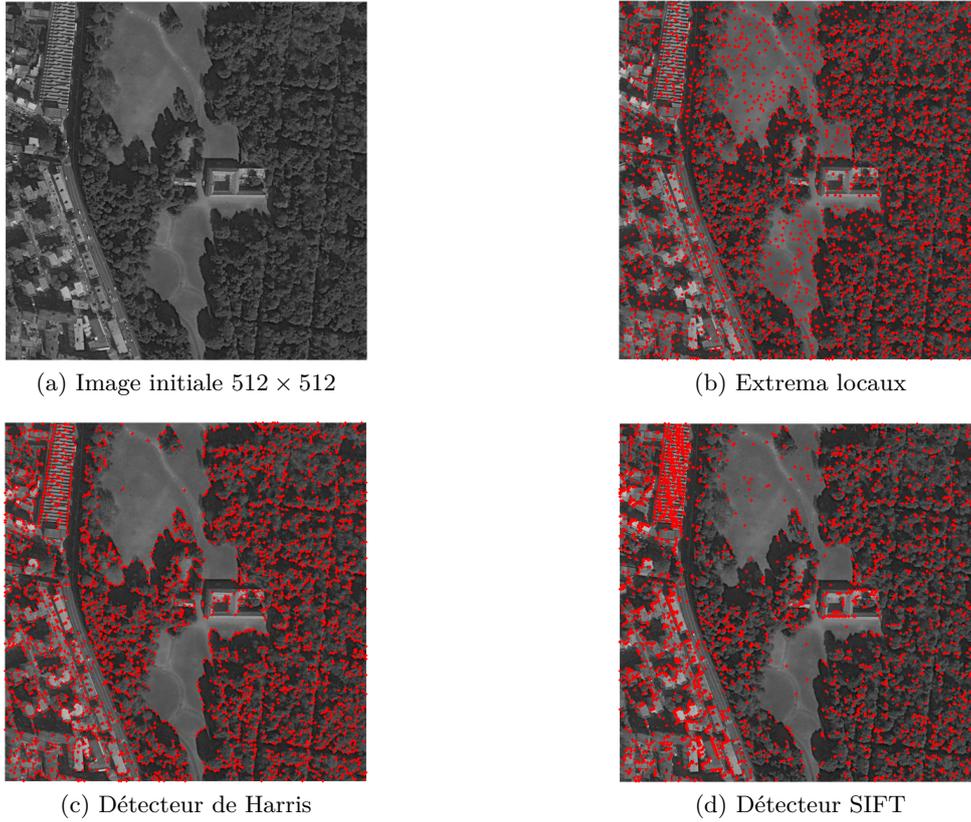


Figure 3 – Evolution de la distribution des points caractéristiques lorsqu'ils sont issus des extrema locaux (a), du détecteur de coins de Harris (b) et du détecteur SIFT (c). Dans tous ces cas, il y a à peu près 3000 points détectés. L'image Pléiades est à 50cm de résolution ©CNES.

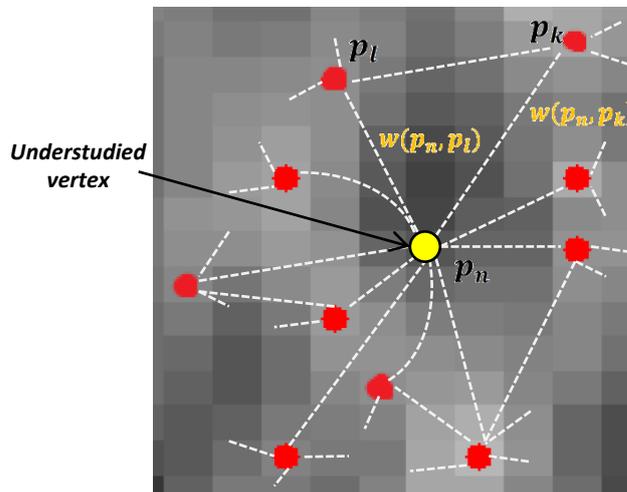


Figure 4 – Construction d'un graphe à partir d'un ensemble de points d'intérêt. Le principe de construction s'applique de façon indifférenciée à un ensemble local de points d'intérêt, fournissant un graph local uniquement, ou à l'ensemble des points d'intérêt de l'image (l'étendue de la mise en relation entre les points d'intérêt doit alors être limitée).

Analyse spectrale par graphe pour la segmentation en texture

La définition de la fonction $\text{dist}(p, q)$ passe tout d'abord par l'extraction d'un vecteur de signature caractéristique de la texture locale entourant un point d'intérêt. La procédure se schématise à travers la figure 5 ci-dessous.

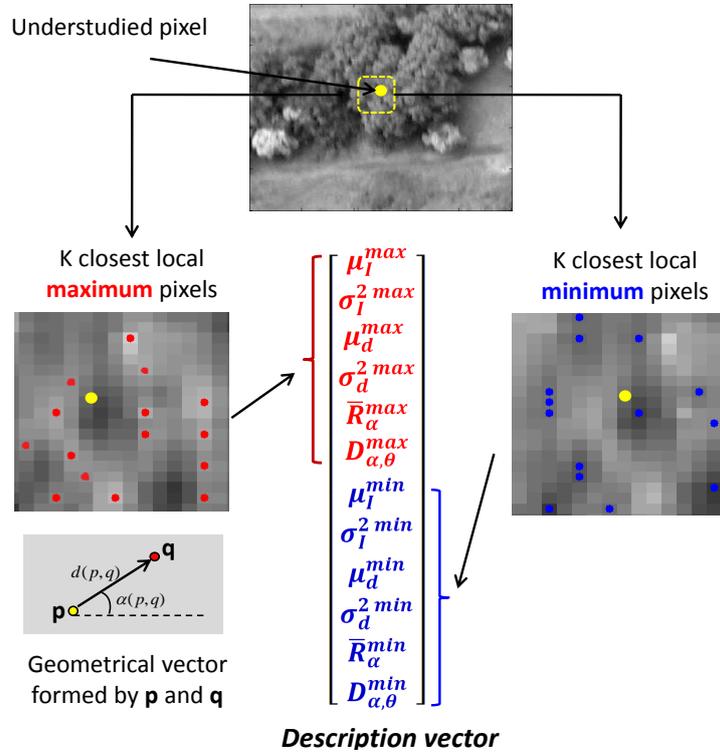


Figure 5 – Vecteur de description proposé pour l'analyse ponctuelle de la texture.

Ce vecteur de signature regroupe plusieurs valeurs caractérisant l'environnement du point clé p et de son entourage à travers ses K plus proches voisins ou de ses voisins à travers une fenêtre d'analyse de taille $W \times W$:

- Moyenne et variance des intensités :

$$\mu_I^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} I(q), \quad (4)$$

$$\sigma_I^2 \max(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (I(q) - \mu_I^{\max}(p))^2. \quad (5)$$

- Moyenne et variance des distances spatiales :

$$\mu_d^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} d(p, q), \quad (6)$$

$$\sigma_d^2 \max(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (d(p, q) - \mu_d^{\max}(p))^2, \quad (7)$$

où $d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$.

- Mesure de la dispersion angulaire et de la concentration directionnelle (issue des statistiques directionnelles [29]) :

$$D_{\alpha,\theta}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \{1 - \cos(\alpha(p,q) - \theta)\}, \quad (8)$$

$$\overline{R}_{\alpha}^{\max}(p) = \sqrt{\overline{C}_{\alpha}^2(p) + \overline{S}_{\alpha}^2(p)}, \quad (9)$$

où

$$\overline{C}_{\alpha}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \cos \alpha(p,q),$$

$$\overline{S}_{\alpha}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \sin \alpha(p,q),$$

$$\alpha(p,q) = \arctan \left(\frac{y_q - y_p}{x_q - x_p} \right), \quad \alpha(p,q) \in [-\pi, \pi].$$

Le vecteur de signature ainsi constitué pour les points clé représentant les maxima locaux et les minima locaux est donc de dimension 12 :

$$\delta^{\text{PW}}(p) = \left[\mu_I^{\max}(p), \sigma_I^2{}^{\max}(p), \mu_d^{\max}(p), \sigma_d^2{}^{\max}(p), \overline{R}_{\alpha}^{\max}(p), D_{\alpha,\theta}^{\max}(p), \right. \\ \left. \mu_I^{\min}(p), \sigma_I^2{}^{\min}(p), \mu_d^{\min}(p), \sigma_d^2{}^{\min}(p), \overline{R}_{\alpha}^{\min}(p), D_{\alpha,\theta}^{\min}(p) \right]. \quad (10)$$

La construction du graphe se fonde donc sur ce vecteur de signature pour définir les poids des connections des nœuds voisins :

$$w(p,q) = e^{-\|\delta^{\text{PW}}(p) - \delta^{\text{PW}}(q)\|_2^2}, \forall (p,q) \in E. \quad (11)$$

Pour être complet, il convient de distinguer le cas des images panchromatiques (où les pixels ont des valeurs scalaires) et les images multispectrales où les pixels sont des vecteurs (signatures spectrales). Dans le second cas, le traitement successif des différentes bandes spectrales conduit à un vecteur de signature de dimension trop importante, c'est pourquoi une analyse en composante principale (ACP) est appliquée tout d'abord à l'image multispectrale. La première bande sert à définir les points clés (par les extrema locaux) ainsi que la première partie du vecteur de signature, puis la deuxième bande spectrale est également utilisée pour fournir des informations spectrales complémentaires afin de compléter ce vecteur de signature qui est à présent de dimension 16.

La classification de l'image repose quant à elle sur deux algorithmes de traitement du signal sur graphe : la classification spectrale (Spectral clustering) qui repose sur la classification des vecteurs propres de la matrice Laplacienne du graphe ainsi construit ; ou la classification par un simple K-moyenne des coefficients en ondelettes du signal sur son graphe.

Les résultats de la figure 6, dont l'analyse est détaillée dans la partie écrite en anglais, montrent tout l'intérêt de cette approche.

Malheureusement, cette méthode se révèle très complexe, notamment à cause du calcul des distances entre les vecteurs de signature des nœuds et également de l'analyse spectrale de la matrice d'adjacence du graphe, dont la taille est directement liée au nombre de points clés extraits de l'image...

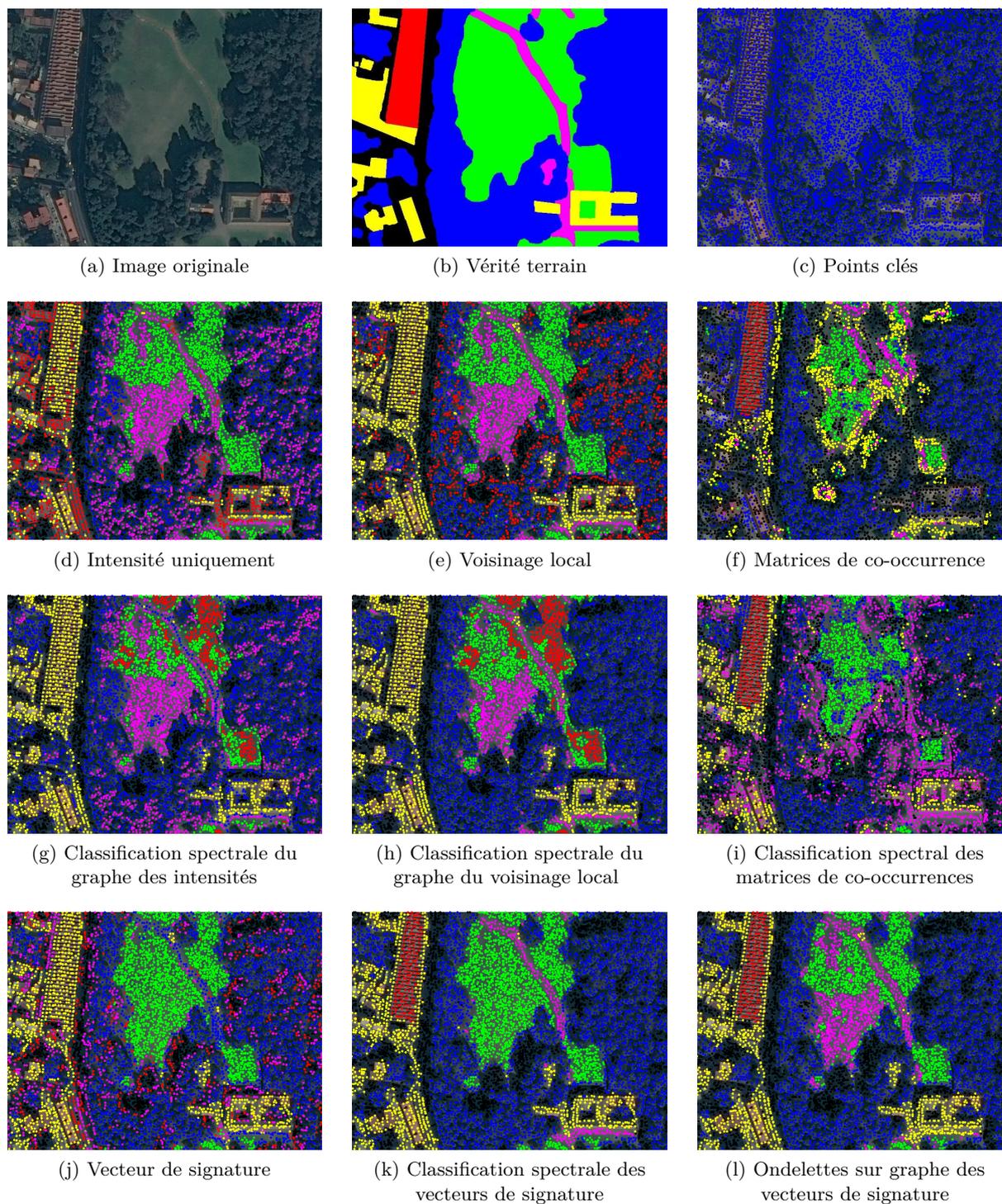


Figure 6 – Résultats de classification de la base n° 1 représentée par une image Pléiades de Naples acquise en février 2013 à 50cm de résolution (sous-région de 600×800 pixels).

Analyse ponctuelle de texture pour l'analyse des changements et des déplacements

Nous nous intéressons ici à l'intérêt qu'apporte une approche ponctuelle dans la détection de changements ou la mesure de déplacement entre deux images. Ici, ce sont des images radar qui sont utilisées dans les illustrations.

Revenons à la construction du graphe dans la représentation des textures par la méthode ponctuelle. Le poids des arêtes est ici fixé de façon relativement classique en considérant la différence d'intensité entre les pixels d'intérêt :

$$w(p, q) = e^{-\gamma d_I(p, q)} = e^{-\gamma \left| \log \frac{\mu_I(p)}{\mu_I(q)} \right|}, \quad (12)$$

où $\mu_I(p)$ est la moyenne locale de l'image I autour du point p (à travers une fenêtre de taille $\omega_p \times \omega_p$). On remarque qu'avec l'utilisation de la valeur absolue, les pondérations sont symétriques entre p et q .

Cette pondération est une version "radarisée" de l'approche par base radiale de la distance entre deux pixels où $d_I(p, q)$ est définie par :

$$d_I(p, q) = |I(p) - I(q)|^2. \quad (13)$$

On obtient donc la matrice d'adjacence \mathcal{W}_I qui caractérise le graphe construit à partir de l'image I et de ses points d'intérêt.

Le principe de la mesure de changement est de valider le fait qu'un signal soit conforme à sa diffusion sur le graphe... Comme on utilise une image SAR, plutôt que de considérer les valeurs de pixels, nous considérons le signal filtré en transformé en σ_0 (grossièrement). Le signal d'une image appliqué aux nœuds d'un graphe est donc :

$$f_I = \left[\log \mu_I(p_1), \log \mu_I(p_2), \dots, \log \mu_I(p_N) \right]. \quad (14)$$

Dans une approche de détection de changements *naïve* fondée sur le rapport de moyennes locales, la mesure de changement s'écrit avec l'équation (15) (ici restreinte aux points d'intérêt).

$$MR_{I_1, I_2}(p) = \left| \log \frac{\mu_{I_1}(p)}{\mu_{I_2}(p)} \right| = |f_{I_1}(p) - f_{I_2}(p)|. \quad (15)$$

Nous utilisons une mesure similaire pour mesurer la différence entre les 2 signaux après diffusion sur le (même) graphe, de matrice d'adjacence \mathcal{W} :

$$GMR_{\mathcal{W}, I_1 | I_2}(p) = MR_{\mathcal{W}f_{I_1} - \mathcal{W}f_{I_2}}(p) = \sum_{q \in \mathcal{N}_{\mathcal{W}}(p)} \omega(p, q) \left| \log \frac{\mu_{I_1}(p)}{\mu_{I_2}(p)} \right|. \quad (16)$$

À l'instar de la mesure de Kullback-Leibler pour la détection de changement [30], la mesure $GMR_{\mathcal{W}, I_1 | I_2}(p)$ n'est pas symétrique entre les 2 images, puisque le graphe est construit avec la localisation de points d'intérêt d'une des deux images. Pour ce faire, il suffit de construire 2 graphes et de voir si la diffusion de 2 images est toujours conforme :

$$GMR_{\mathcal{W}, I_1, I_2}(p) = GMR_{\mathcal{W}_{I_1}, I_1 | I_2}(p) + GMR_{\mathcal{W}_{I_2}, I_2 | I_1}(p). \quad (17)$$

Il est intéressant de noter ici que le graphe est en fait implicite. Il n'est, en effet, pas nécessaire de construire de graphe en tant que tel avant d'évaluer la mesure de similarité. L'éq. (17) peut être évaluée directement sans recours à la notion de graphe. Cela a son importance lorsque l'on traite de grosses images puisqu'il n'y a pas à gérer l'occupation mémoire de la matrice d'adjacence mais uniquement les distances locales entre les points p et q au fur et à mesure.

Nous avons appliqué cette mesure de détection de changements à un couple d'images Radarsat-1 (en mode F2 et F5) acquises avant et après l'éruption du volcan Nyiragongo en R.D. du Congo en janvier 2002.

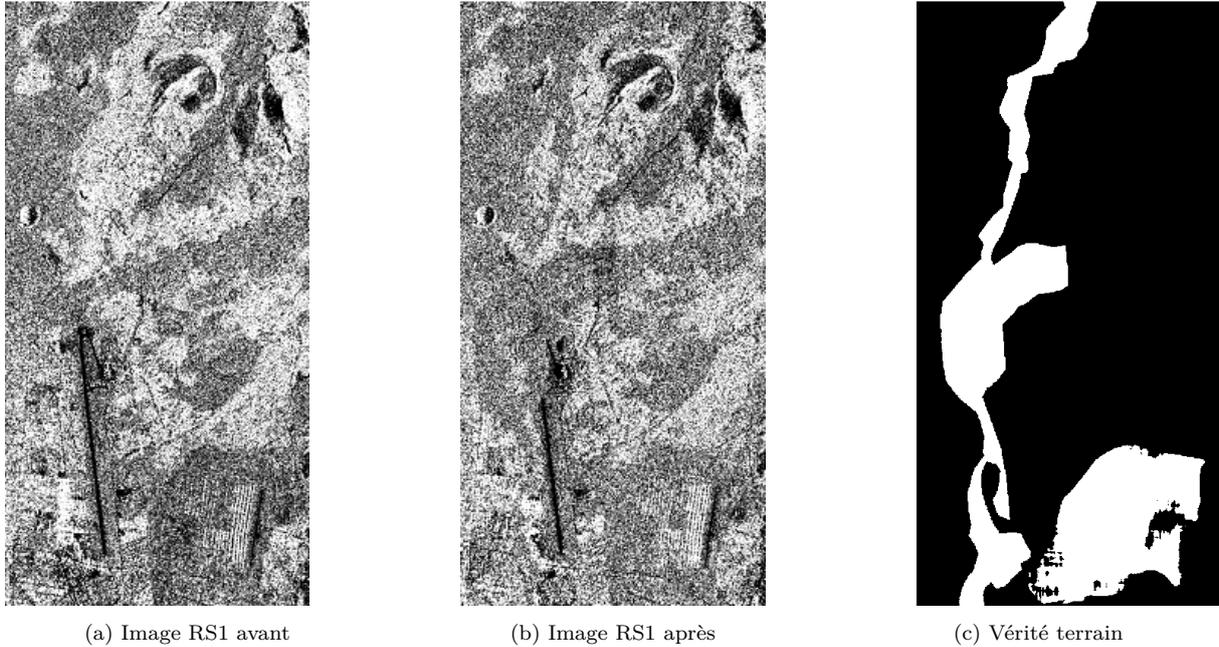


Figure 7 – Images Radarsat-1 acquises avant et après l’éruption du volcan Nyiragongo en janvier 2002.

Les données initiales sont présentées à la figure 7 et le résultat issu de la mesure de l’équation (17) est présenté à la figure 8-(e). Sur cette figure 8, plusieurs mesures standardes ont été utilisées pour comparaison (dans leur restriction aux points d’intérêt). Il s’agit des mesures classiques que l’on peut trouver dans la littérature. Il s’agit du rapport des moyennes locales, sa version log, la mesure de Kullback-Leibler avec hypothèse gaussienne des distributions locales ou via un développement de Edgeworth des histogrammes locaux [30]. Toutes ces mesures utilisent une fenêtre glissante pour estimer leurs paramètres internes, toutefois, nous n’avons appliqué le calcul de la mesure de changements qu’aux points d’intérêt pour avoir un résultat homogène et comparable avec notre méthode par graphe. L’affichage des images de la figure 8 est en couleur avec des niveaux déterminés par un simple K-means à 6 niveaux représentés entre le bleu (non changement) et le rouge (changement sûr) avec l’ordonnancement suivant : bleu < vert < cyan < jaune < magenta < rouge.

Plutôt que d’appliquer un K-means à 6 niveaux, il est possible de faire un simple seuil et de compter, à l’aide du masque, les bonnes détections et les fausses alarmes. On obtient alors les courbes de performances de la figure 9. Il apparaît clairement que la mesure par graphe est plus performante que les autres mesures en utilisant d’ailleurs un nombre de points plus restreint dans l’extraction des paramètres.

Il est intéressant de remarquer que l’approche proposée apparaît comme une intermédiaire entre les méthodes classiques fondées sur l’utilisation de fenêtre glissante, et les approches non-locales fondées sur la recherche de similarité entre situations locales avant de procéder à la mesure de similarité. La figure 10 illustre ces différentes stratégies. Si les méthodes par fenêtres glissantes ont une complexité réduite en $O(W^2)$, W étant la taille de la fenêtre de recherche, les approches non-locales ont une complexité beaucoup plus importante en $O(W^2 w_p^2)$, w_p étant la taille du patch. La méthode que nous proposons ici présente une complexité intermédiaire en $O(K w_p^2)$ où K est le nombre de voisin (i.e. $K \ll W^2$), pour une performance déjà supérieure.

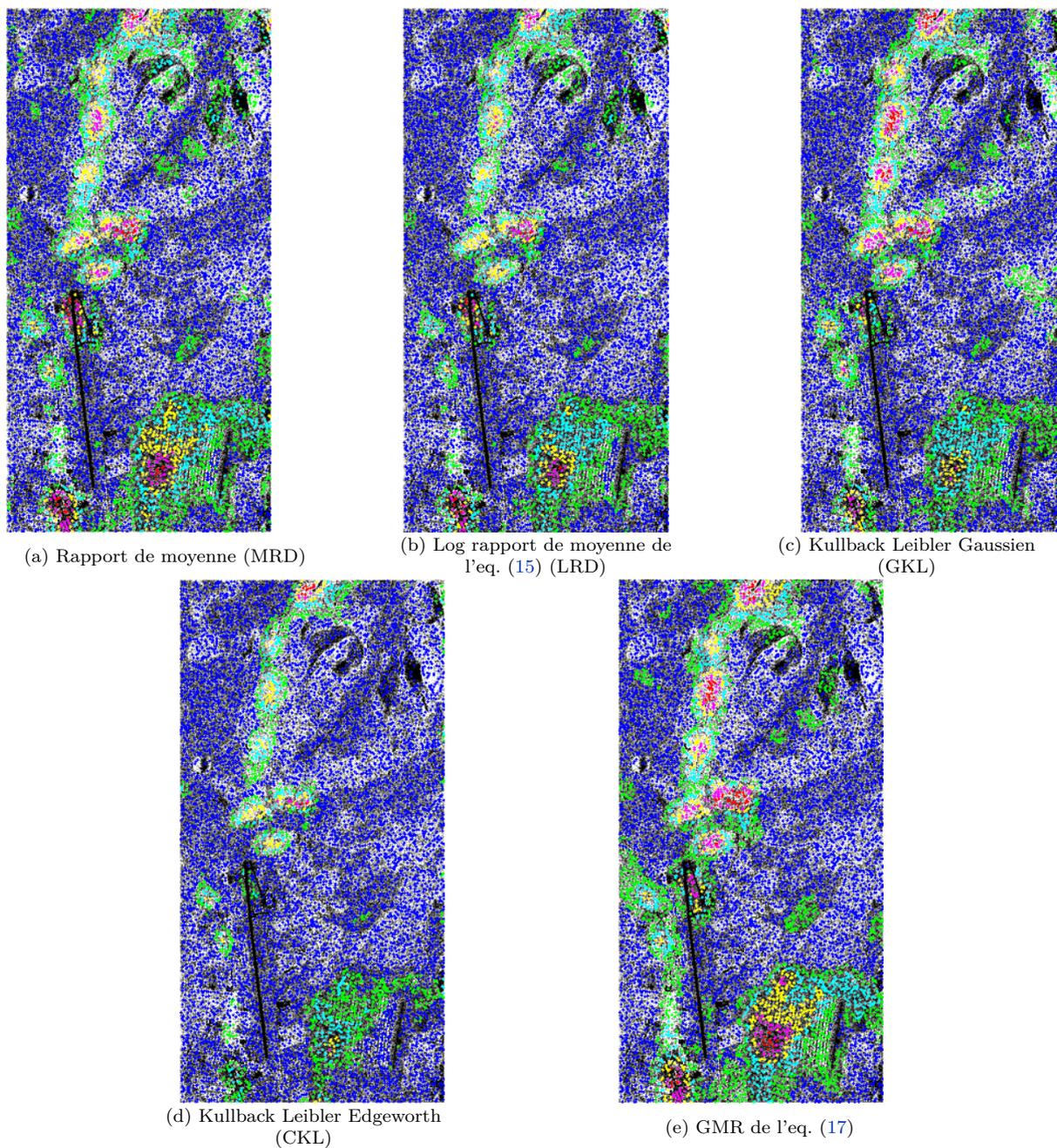


Figure 8 – Mesure de changements appliquée aux images avant/après de Goma, en R.D. Congo.

La mesure de détection de changements ainsi décrite peut également être utilisée pour corrélérer des textures et mesurer un déplacement entre deux images. C'est ce que nous avons expérimenté dans cadre de l'estimation de déplacements de glacier à partir d'images TerraSAR-X.

Le principe général est résumé à la figure 11. La mesure de changement, fondée sur l'adéquation d'un signal sur un graphe construit à partir d'un autre signal, est appliqué localement en construisant un graphe à partir d'une image Maître et d'évaluer la mesure de changement en utilisant l'image Esclave décalée d'un vecteur de déplacement. On cherche le vecteur de déplacement qui minimise la mesure de changement...

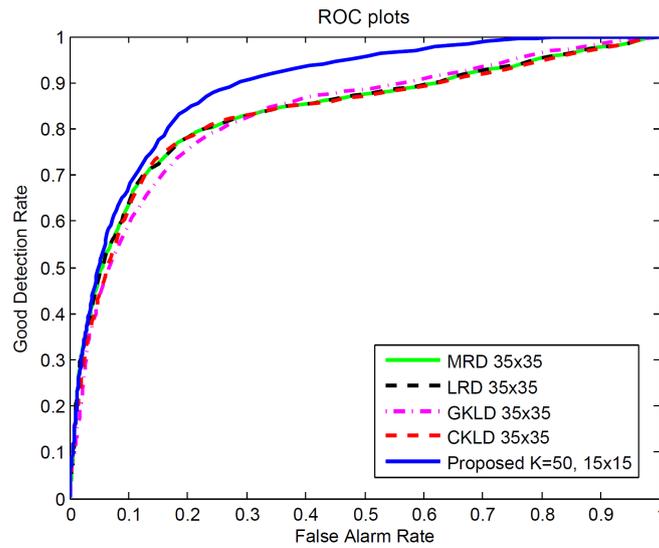


Figure 9 – Courbe de performance des mesures de changement présentées à la figure 8.

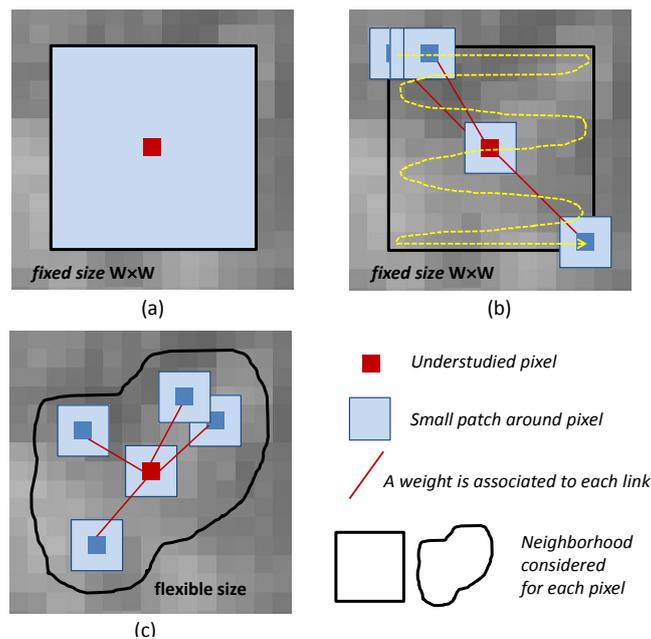


Figure 10 – Différents modes de gestion du voisinage et de la similarité entre les situations locales : (a) analyse par fenêtre glissante, (b) approche non-locale, (c) approche proposée par points clés définis avant la mesure de similarité.

Les résultats obtenus sont finalement conformes à ceux que l'on peut retrouver dans l'état de l'art consacré aux mesures de déplacement de glacier.

C'est lorsque la texture est homogène et peu structurée (à l'instar d'un bruit blanc) que notre méthode montre le plus d'aberrations. En cela, nous ne sommes pas meilleurs que l'état de l'art...

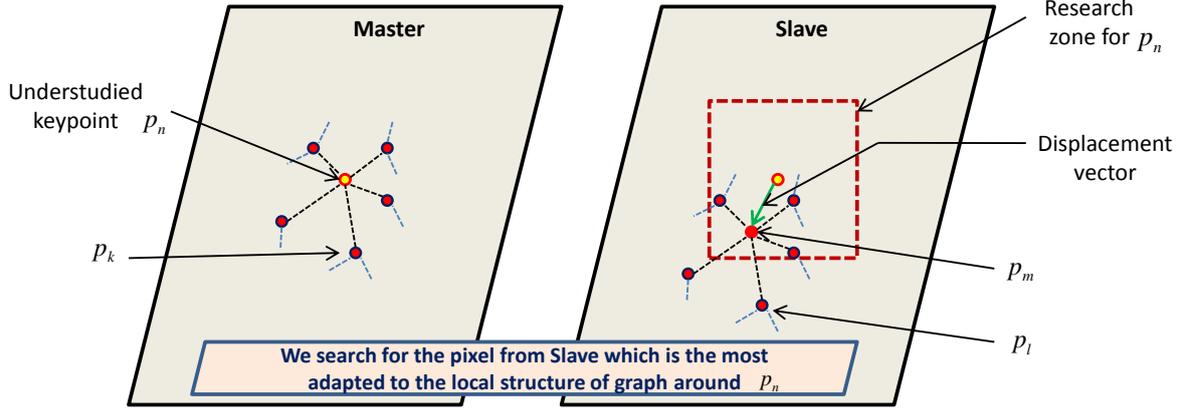


Figure 11 – Principe général de corrélation de texture par approche ponctuelle.

Approche ponctuelle et texture structurée

Nous nous intéressons ici à la façon dont nous pouvons introduire des informations d'orientation de texture dans notre vecteur de signature.

Covariance locale sur points clés

L'ajout d'information structurelle dans le vecteur de signature peut se faire en s'inspirant des notions de tenseurs de structure [31, 32]. Le vecteur de signature est alors défini par le vecteur suivant à 6 composantes :

$$f_p \in \mathbb{R}^6 \quad / \quad f_p = [I_p, I_p^x, I_p^y, I_p^{xx}, I_p^{yy}, I_p^{xy}]^T, \quad (18)$$

I_p représente l'intensité du pixel au point clé p ; I_p^x (resp. I_p^y) représente le gradient horizontal (resp. vertical) de l'image au point p ; I_p^{xx} , I_p^{yy} , I_p^{xy} représentent les dérivées secondes.

La mesure de distance entre des vecteurs de signature va à présent se fonder sur la mesure de similarité des covariances :

$$C_p = \frac{1}{W^2} \sum_{q \in \mathcal{N}_W(p)} (f_q - \mu_p)(f_q - \mu_p)^T, \quad (19)$$

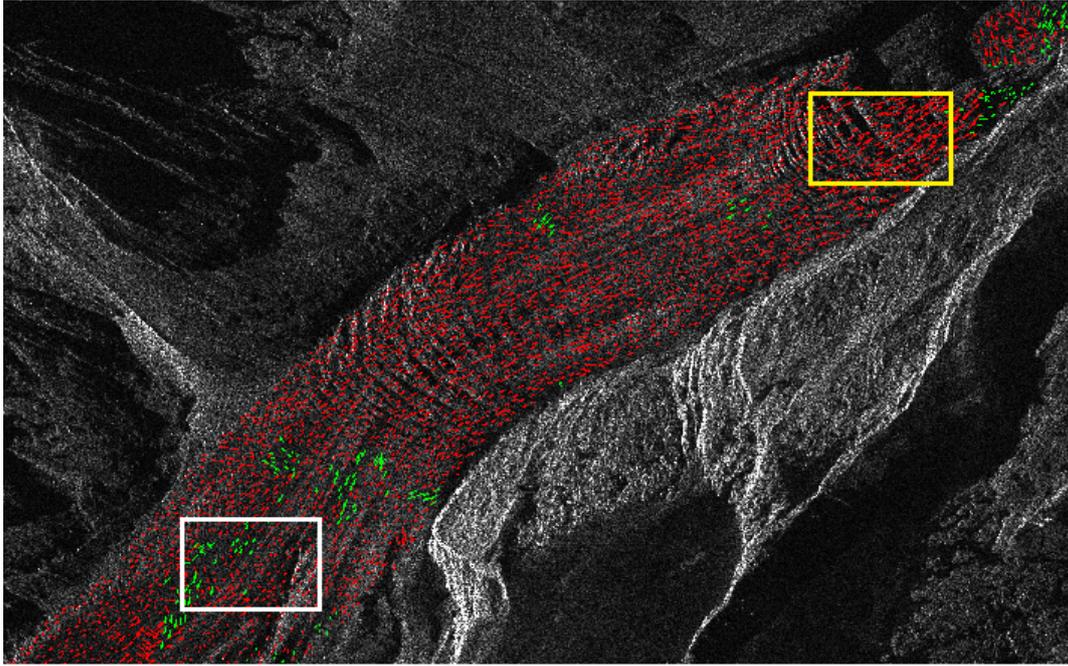
où μ_p est le vecteur moyen estimé dans un voisinage (ici contigu) autour de p : $\mathcal{N}_W(p)$,

$$\mu_p = \frac{1}{W^2} \sum_{q \in \mathcal{N}_W(p)} f_q. \quad (20)$$

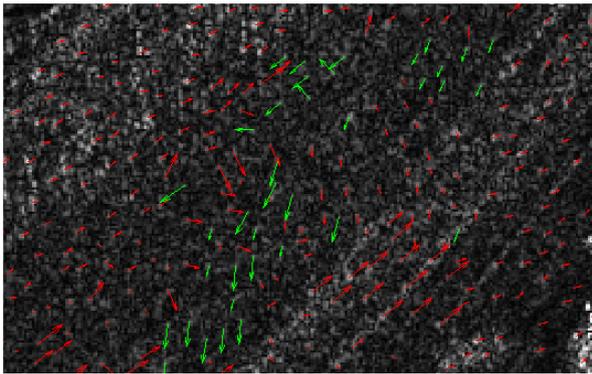
Finalement, le vecteur de signature est fondé sur les mesures de covariance des vecteurs f_q à 6 composantes, restreint aux points clés représentant les maxima locaux et les minima locaux :

$$C_p^{\max} = \frac{1}{|\mathcal{N}_W^{\max}(p)|} \sum_{q \in \mathcal{N}_W^{\max}(p)} (f_q - \mu_p^{\max})(f_q - \mu_p^{\max})^T, \quad (21)$$

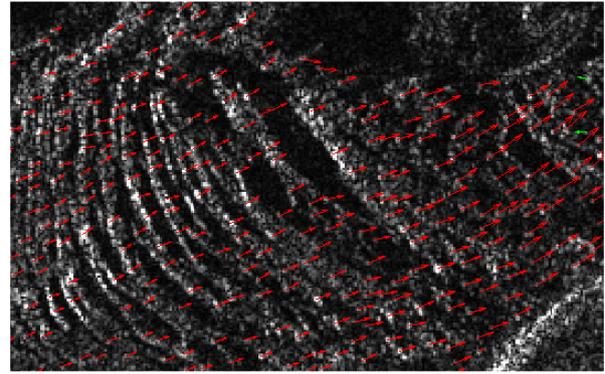
$$C_p^{\min} = \frac{1}{|\mathcal{N}_W^{\min}(p)|} \sum_{q \in \mathcal{N}_W^{\min}(p)} (f_q - \mu_p^{\min})(f_q - \mu_p^{\min})^T. \quad (22)$$



(a) Mesure de déplacement



(b) Zoom sur le carré blanc



(c) Zoom sur le carré jaune

Figure 12 – Mesure de déplacement des glaciers issue de notre approche fondée sur l’adéquation d’un signal à un graphe construit à partir d’une autre image. Les zones où la texture est structurée donnent des résultats satisfaisants mais là où la texture est isotrope et s’apparente à un signal homogène, la mesure présente beaucoup d’aberrations.

La mesure de distance entre ces matrices de covariance doit prendre en compte la variété sur laquelle évoluent ces matrices de covariance. Ainsi, la distance Riemannienne a été utilisée :

$$\rho(C_1, C_2) = \sqrt{\sum_{\ell=1}^d \ln^2 \lambda_{\ell}}. \quad (23)$$

Elle se fonde sur l’évaluation des valeurs propres généralisées entre les deux matrices C_1 et C_2 :

$$\lambda_{\ell} C_1 \chi_{\ell} - C_2 \chi_{\ell} = 0, \quad \ell = 1, \dots, d, \quad (24)$$

d étant la dimension des matrices de covariance.

Comme cette distance est appliquée aux covariances des maxima locaux et des minima locaux sans interférence, la distance Riemannienne est appliquée deux fois :

$$\text{dist} \left[\delta^{\text{PW-COG}}(p_1), \delta^{\text{PW-COG}}(p_2) \right] = \rho(C_{p_1}^{\text{max}}, C_{p_2}^{\text{max}}) + \rho(C_{p_1}^{\text{min}}, C_{p_2}^{\text{min}}). \quad (25)$$

Pour la classification des images, un simple K-moyenne est appliqué en utilisant cette distance de l'éq. (25).

La figure 13 illustre la performance de ce point de vue. On y voit l'intérêt d'adopter le

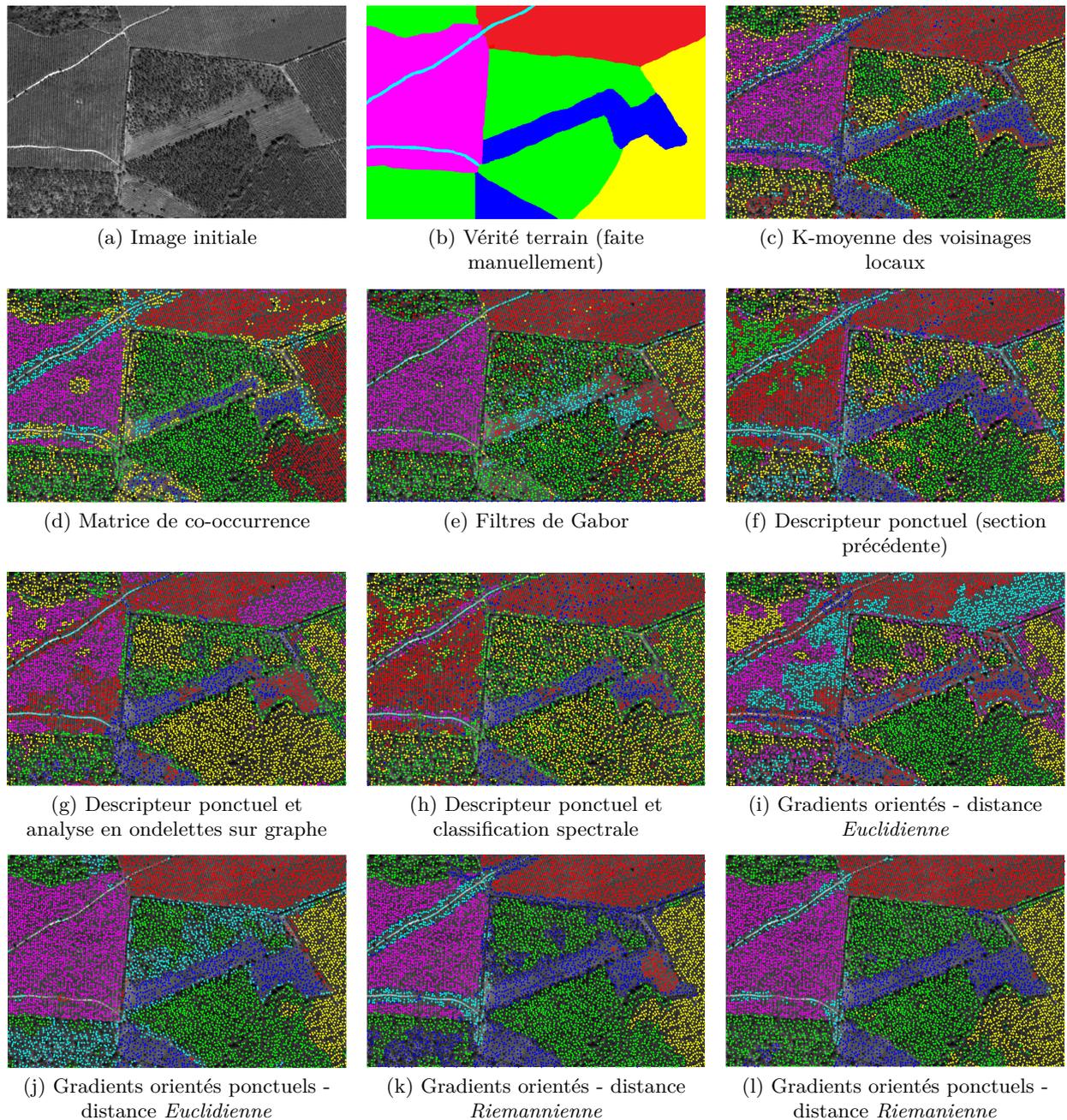


Figure 13 – Résultats de classification non-supervisée sur une image Pléiades acquise en juin 2012 entre Bordeaux et Arcachon dans une zone viticole mixte. L'approche ponctuelle des gradients orientés montre ici tout son intérêt.

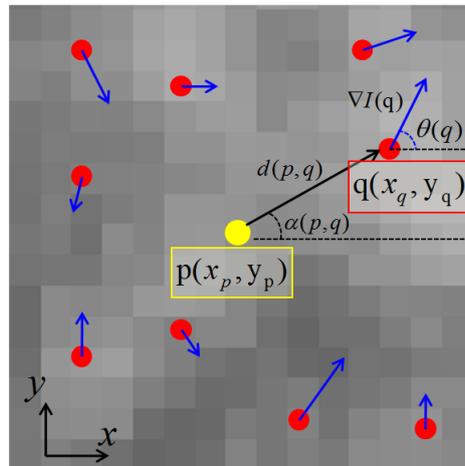


Figure 14 – Information géométrique et gradient local des points clés représentant un maximum local (resp. minimum local) $q = (x_q, y_q)$ à travers l'ensemble $S_K^{\max}(p)$ (resp. $S_K^{\min}(p)$) considérés pour évaluer le vecteur de signature du point clé $p = (x_p, y_p)$.

point de vue des gradients orientés dans la caractérisation des textures mais aussi d'utiliser la distance Riemanienne dans l'évaluation des distances entre matrices de covariance. De plus, la complexité algorithmique de cette approche est bien plus intéressante que les approches par graphes puisque quelques secondes suffisent à traiter cette image 800×1300 pixels. En outre, une extension supervisée de cet algorithme peut être proposée et est abordée dans la partie anglaise.

Descripteur local sur points clés

Finalement, nous aboutissons à la définition la plus aboutie des vecteurs de signature permettant de caractériser des textures, qu'elles soient orientés ou non...

L'algorithme général reste inchangé : des points clés sont extraits par critères de maxima et minima locaux. Pour chaque point d'intérêt p , on extrait son vecteur de signature en fonction de ses K voisins (de même type min ou max local que p). Ce vecteur de signature est composé des mesures suivantes, selon le schéma de la figure 14.

- Moyenne et variance des intensités des extrema, cf. éq. (4) et (5) :

$$\mu_I^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} I(q), \quad (26)$$

$$\sigma_I^2{}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (I(q) - \mu_I^{\max}(p))^2. \quad (27)$$

- Moyenne et variance des distances spatiales des points clés, cf. éq. (6) et (7) :

$$\mu_d^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} d(p, q), \quad (28)$$

$$\sigma_d^2{}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (d(p, q) - \mu_d^{\max}(p))^2, \quad (29)$$

où $d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$.

- Mesure statistique directionnelle des points clés dans le voisinage de p :

$$\sigma_{\text{cir},\alpha}^{2\text{max}}(p) = 1 - \sqrt{\overline{C}_\alpha^2(p) + \overline{S}_\alpha^2(p)}, \quad (30)$$

où

$$\overline{C}_\alpha(p) = \frac{1}{K} \sum_{q \in S_K^{\text{max}}(p)} \cos \alpha(p, q),$$

$$\overline{S}_\alpha(p) = \frac{1}{K} \sum_{q \in S_K^{\text{max}}(p)} \sin \alpha(p, q),$$

$$\alpha(p, q) = \arctan \left(\frac{y_q - y_p}{x_q - x_p} \right), \quad \alpha(p, q) \in [-\pi, \pi], \forall p, q.$$

- Moyenne et variance des gradients (en magnitude) de chaque point clé q autour de p :

$$\mu_{|\nabla I|}^{\text{max}}(p) = \frac{1}{K} \sum_{q \in S_K^{\text{max}}(p)} |\nabla I(q)|, \quad (31)$$

$$\sigma_{|\nabla I|}^{2\text{max}}(p) = \frac{1}{K} \sum_{q \in S_K^{\text{max}}(p)} (|\nabla I(q)| - \mu_{|\nabla I|}^{\text{max}}(p))^2, \quad (32)$$

où $|\nabla I|$ est le magnitude du gradient fourni par l'application du filtre de Sobel.

- Statistiques directionnelles des gradients des points clés q :

$$\sigma_{\text{cir},\theta}^{2\text{max}}(p) = 1 - \sqrt{\overline{C}_\theta^2(p) + \overline{S}_\theta^2(p)}, \quad (33)$$

avec

$$\overline{C}_\theta(p) = \frac{1}{K} \sum_{q \in S_K^{\text{max}}(p)} \cos \theta(q),$$

$$\overline{S}_\theta(p) = \frac{1}{K} \sum_{q \in S_K^{\text{max}}(p)} \sin \theta(q),$$

où θ est l'orientation du vecteur gradient ∇I .

On a donc $\delta^{\text{max}}(p)$ le vecteur de caractéristique de p en prenant en compte ses K points clés voisin de type maximum local comme p .

$$\delta^{\text{max}}(p) = \left[\mu_I^{\text{max}}(p), \sigma_I^{2\text{max}}(p), \mu_d^{\text{max}}(p), \sigma_d^{2\text{max}}(p), \sigma_{\text{cir},\alpha}^{2\text{max}}(p), \mu_{|\nabla I|}^{\text{max}}(p), \sigma_{|\nabla I|}^{2\text{max}}(p), \sigma_{\text{cir},\theta}^{2\text{max}}(p) \right] \in \mathbb{R}^8. \quad (34)$$

On définit $\delta^{\text{min}}(p)$ de façon équivalente en prenant en compte les K points clés autour de p qui représentent des minima locaux.

Finalement, le vecteur de signature de p regroupe l'intensité du pixel p , et des vecteurs de caractéristiques $\delta^{\text{max}}(p)$ et $\delta^{\text{min}}(p)$:

$$\delta^{\text{LED}}(p) = \left[I(p), \delta^{\text{max}}(p), \delta^{\text{min}}(p) \right] \in \mathbb{R}^{17}. \quad (35)$$

Le critère de distance utilisé pour évaluer la similarité entre les vecteurs de signature est fondée sur la distance Riemannienne entre les matrices de covariance issues de $\delta^{\text{LED}}(p)$:

$$C_i = \frac{1}{N_i} \sum_{p \in S_i} \left(\delta^{\text{LED}}(p) - \mu_i \right)^T \left(\delta^{\text{LED}}(p) - \mu_i \right), \quad (36)$$

avec

$$\mu_i = \frac{1}{N_i} \sum_{p \in S_i} \delta^{\text{LED}}(p). \quad (37)$$

$$d_{\text{riemannienne}}(I_i, I_j) = \sqrt{\sum_{\ell=1}^d \ln^2 \lambda_\ell} \quad (38)$$

venant des valeurs propres généralisées de $\lambda_\ell C_i \chi_\ell - C_j \chi_\ell = 0$, $\ell = 1, \dots, d$. Pour les expérimentations, nous avons également évalué les performances de la distance de Mahalanobis :

$$d_{\text{mahalanobis}}(I_i, I_j) = (\mu_i - \mu_j) \left(C_i^{-1} + C_j^{-1} \right) (\mu_i - \mu_j)^T. \quad (39)$$

Cet algorithme a été appliqué à des données Pléiades acquises dans le cadre de la caractérisation des parcelles viticoles dans les domaines de Pessac-Léognan et Saint-Emilion. Dans ce contexte supervisé, nous disposons d'une base d'images de types de sol synthétisés dans le tableau 1.

Table 1 – Nombre de patches utilisés pour la reconnaissance supervisée de type d'utilisation des sols.

Base	Forêt	Sol nu	Zone urbaine	Parcelles viticoles	Total
<i>Pessac 22-08-2012</i>	66	53	147	179	445
<i>Emilion 03-09-2013</i>	44	32	27	881	984

Les résultats regroupés dans le tableau 2, et qui sont plus longuement commentés dans la version anglaise, montrent tout l'intérêt de notre approche.

Table 2 – Performance de classification supervisée, évaluée en taux de reconnaissance moyen (%).

Methode	Pessac 22-08-2012	Emilion 03-09-2013
Modèle Gaussien	77.51	78.35
SIRV	60.58	60.16
Copule Gaussienne	76.88	75.91
Matrice de Co-occurrence	54.56	64.57
Gabor	61.37	62.39
Descripteur de Weber	64.38	73.88
Descripteur ponctuel PW	78.24	83.18
LED (K=15,dist. Mahalanobis)	83.42	87.90
LED (K=20,dist. Mahalanobis)	83.78	88.18
LED (K=15,dist. Riemannienne)	85.63	89.01
LED (K=20,dist. Riemannienne)	85.79	89.43

La reconnaissance des parcelles viticoles elle-même peut-être illustrée avec la figure 15.

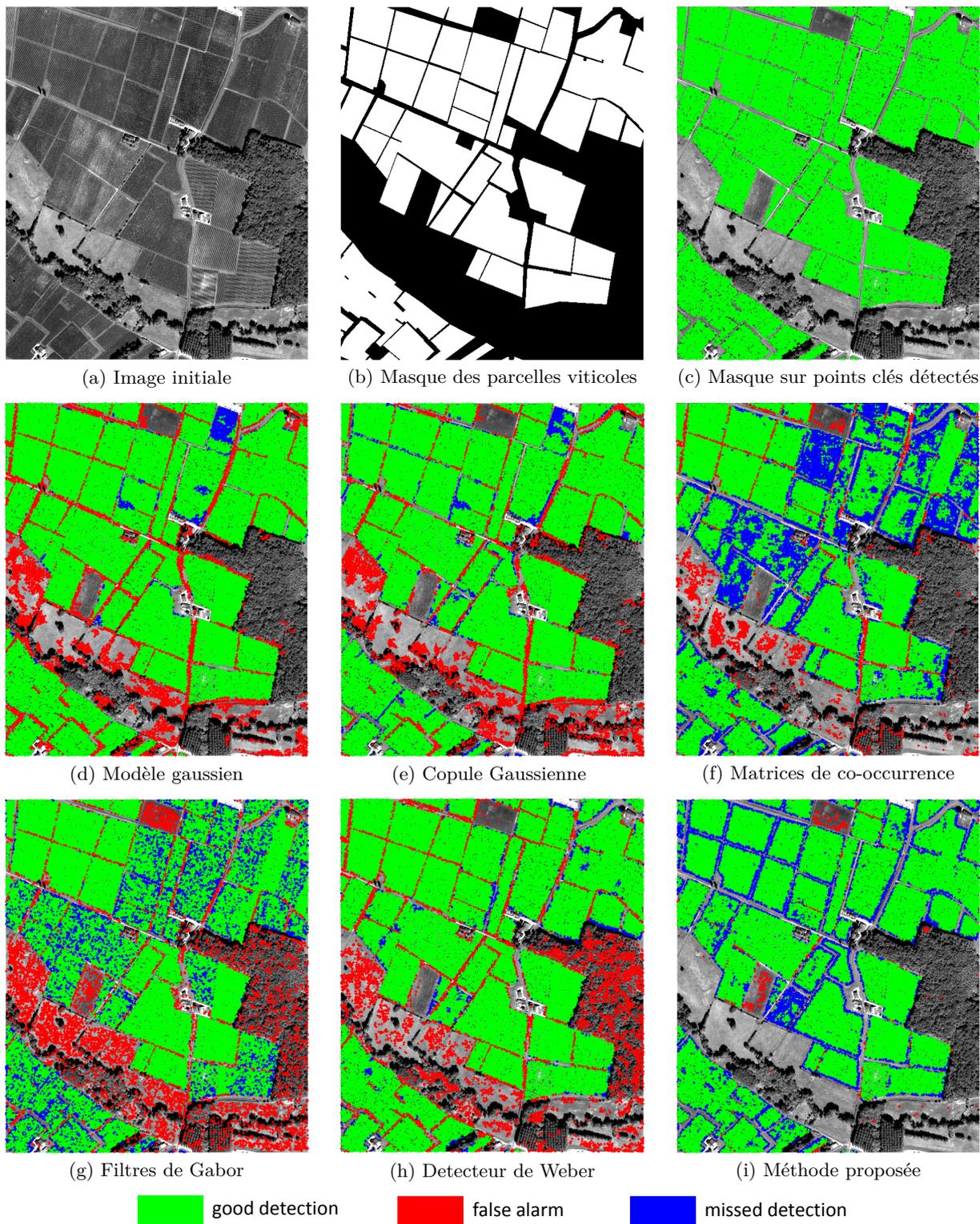


Figure 15 – Détection des parcelles viticoles dans le domaine de Saint Emilion à partir d’une image Pléiades de 2000×1700 pixels.

Conclusion et perspectives

Dans ce travail de thèse, nous avons développé une méthode d'analyse de texture qui est tout d'abord fondée sur une approche ponctuelle avec laquelle, seuls des points clés, ici représentés par des extrema locaux, suffisent à appréhender le signal. Ensuite, un vecteur de signature qui permet de caractériser le voisinage de chaque point clé en utilisant des critères d'intensité, de distance spatiale, de statistique directionnelle et également des critères liés au gradient de ces points caractéristiques. Finalement, la mesure de distance permettant d'évaluer la similarité entre les points clés de l'image est fondée sur une distance riemannienne entre les matrices de covariance construite à partir des vecteurs de signature des points clés.

Cette approche a été appliquée avec succès à des images optiques satellitaires de très haute résolution spatiale, qu'elles soient panchromatiques ou multispectrales, dans un contexte non supervisé ou supervisé...

De plus, nous avons également proposé une méthode ponctuelle pour le suivi de texture et l'appliquée pour l'analyse des changements et des déplacements de glacier utilisant des données SAR. Les résultats obtenus nous montrent l'efficacité de la méthode comparée avec des approches classiques.

Perspectives

- 1) L'application aux images radar, polarimétriques ou non, pose problème. Les expérimentations qui ont été faites lors de ce travail de thèse nous ont obligé tout d'abord à recourir au filtrage anti-speckle pour une caractérisation plus fiable des points clés. En outre, les résultats d'analyse de texture n'ont pas montré un intérêt thématique pertinent. Il convient donc de remettre en question l'utilisation des points clés à travers la notion d'extrema locaux qui n'est peut-être pas la plus appropriée et de définir un vecteur de signature plus adapté aux textures radar...
- 2) Les résultats de notre approche ponctuelle sont ponctuels... Il conviendrait de revenir à un résultat dense qui correspond à l'utilisation habituelle des cartes d'occupation des sols. Des approches morphologiques ou le recours aux superpixels pourraient nous aider dans la réalisation de ce post-traitement.
- 3) La plupart des méthodes d'analyse de texture de la littérature se fonde sur une analyse multi-échelle de l'image. Il conviendrait donc d'étudier une extension multi-échelle ou multi-résolution de notre approche ponctuelle.
- 4) L'extension de notre approche à l'analyse des images multispectrales est fondée sur l'utilisation de l'ACP pour ne retenir que les deux premières bandes et étendre le vecteur de signature à ces deux bandes... Le recours aux tenseurs de structure, et notamment au gradient de Di Zeno, pourrait éviter le recours à l'ACP et concentrer l'information multi-composante tout en gardant une notion spectrale explicite...
- 5) Les résultats de l'estimation des déplacements de glacier ont montré que dans les zones de texture peu structurée, les mesures ne sont pas fiables. A l'instar des résultats de l'analyse de texture des images radar, il convient de remettre en cause la notion d'extrema locaux dans la définition des points clés et la définition du vecteur de signature pour une meilleure adaptation de notre approche aux textures vues par un capteur radar.

Part I

Introduction

General introduction

The good life is one inspired by love and guided by knowledge.

Bertrand Russell

1.1	Context of the thesis	23
1.2	Classical approaches for texture analysis	24
1.3	Motivation and objective	26
1.4	Thesis overview and organization	27

1.1 Context of the thesis

Exploiting satellite image data to understand and monitor the land cover and land use from the earth surface is one of the most significant task of remote sensing. Texture, together with other features such as spectral signature (or color information), shape, edge, etc., plays an important role in image analysis and interpretation. This thesis is dedicated to the context of texture analysis for very high resolution (VHR) images in the scope of remote sensing imagery. With the ongoing development of both spaceborne and airborne remote sensing systems, the spatial resolution of acquired images has been considerably improved. For instance, some optical satellites can provide a spatial resolution less than 1m such as Pléiades-1A (0.5m), GeoEye-2 (0.34m) or Worldview-3 (0.31m). Then, Synthetic Aperture Radar (SAR) systems such as Radarsat-2 and TerraSAR-X are able to perform their highest resolution up to 1m.

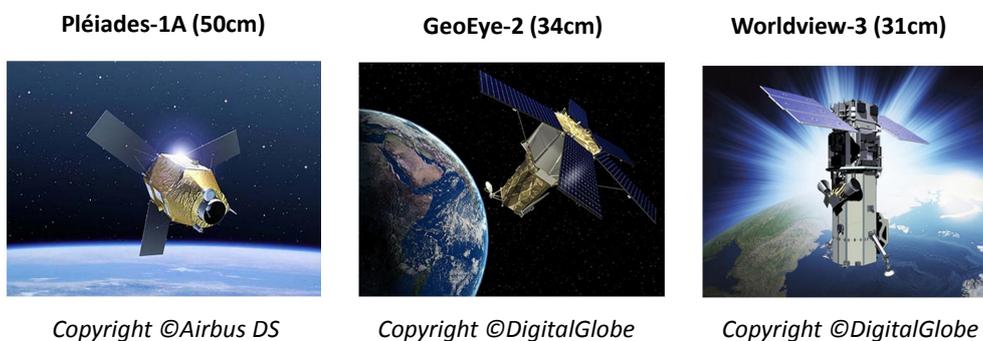


Figure 1.1 – Example of VHR optical satellites.

The increase in spatial resolution of satellite sensors enables us to capture and observe more details and properties of the scene characterized by the acquired image. However, this may pose a challenge for texture analysis task since some areas to be characterized become too small or do not respect the stationarity hypothesis. In fact, within VHR images acquired from optical or SAR sensors, many objects appear quite heterogeneous and their size remains too limited for an appropriate parametric estimation such as in case of different kinds of trees in a mixed forest, vegetation parcels in sub-urban areas, land use in pasture or fallow agricultural fields, etc. Hence, classical methods for texture analysis, which will be briefly reviewed in the next section, may encounter difficulties since they usually assume a stationary or locally stationary hypothesis to estimate their own parameters in order to represent and describe textural features.

Another challenge that can be found in VHR remote sensing imaging is the increase in volume of acquired image data. The more details an image can capture, the more number of pixels it must contain and hence the larger volume of data we need to store and process. This issue may lead to the problem of calculation time of any texture analysis algorithm which proportionally increases with the image size. Therefore, a low-complexity algorithm for texture characterization in VHR images is also an objective of this work.

The next section will provide a brief review of classical methods for texture analysis from which we find that they may be no longer relevant for VHR images. Hence, another strategy is going to be proposed to tackle this task by taking into account the above mentioned challenges. This is the main objective and motivation of this thesis which will be introduced in details in Section 1.3. Finally, Section 1.4 concludes this introduction chapter with the overview and organization of the thesis.



Figure 1.2 – Example of VHR optical image acquired by Pléiades-1A satellite, ©Airbus DS.

1.2 Classical approaches for texture analysis

Many methods have been proposed for texture analysis in the literature. A systematic review can be found in papers [1–3]. Indeed, those methods can be categorized into four principal

approaches including the statistical approach, model-based approach, transform-based approach and structural approach. We now briefly review each of these approaches.

Statistical approach: This approach consists of texture analysis methods based on the statistical distribution of the image pixel's gray-levels. Here, texture is described by a collection of statistic features extracted directly from the image or from the image histogram which represents the distribution of pixel intensities (i.e. intensity histogram). For instance, this approach may be divided into first-order, second-order and higher-order statistics. Some emblematic methods which can be listed here are the local histogram [4], the gray-level cooccurrence matrix (GLCM) [6], the autocorrelation function [5], etc. Among these, the GLCM method appears to be the most popular and commonly used for texture analysis in image processing field. This method is based on the joint probability distribution of pair of pixel's gray-levels. Different types of Haralick features extracted from cooccurrence matrices are exploited to form textural descriptors. For more details about the GLCM method and Haralick features, readers are invited to consult papers [6, 33].

Model-based approach: This can be separated into fractal model and stochastic model methods which attempt to interpret an image texture by the use of generative image model or stochastic model, respectively. Parameters dedicated to the estimated model will be then used to analyze and describe the image. Methods based on fractal model have been shown to be useful for modeling natural texture images. They describe objects having high degree of irregularity in a textural image and provide a measure of statistical quality of image roughness and self-similarity at different scales. More details can be found in [7–9].

For stochastic models, two popular strategies that can be stated here are the auto-regressive (AR) model [10] and the Markov Random Field (MRF) model [11]. The AR model involves the local interaction among image pixels within a neighborhood. It assumes that each pixel intensity is a weighted sum of neighboring pixel intensities according to a vector of model parameters θ which will be then used for texture discrimination [10]. About the MRF model, it works based on the assumption that the intensity of each pixel in a given image depends only on the intensities of its neighboring pixels, not on the whole image. Each pixel is modeled as a site on the lattice from which a texture can be analyzed based on the total energy of the lattice or the interaction among neighboring pixels within the lattice [11].

Transform-based approach: Many transform techniques have been proposed for image texture characterization such as the Fourier transform, the discrete Cosine transform or the wavelet transform, etc. In general, after applying a transform to an input image, the obtained coefficients will be exploited or manipulated in order to construct texture descriptors.

While methods based on the Fourier transform may perform poorly in practice due to its lack of spatial localization, the Gabor filter banks [12, 13], which are based on the windowed Fourier transform, can provide a better spatial localization but still have limitations coming from their non-orthogonal property. The most widely used method within this transform-based approach must be the multiscale image analysis based on wavelet transform. Several statistical parameters or distribution models derived from the space of wavelet coefficients are exploited for texture description task in many research studies [14–17].

Structural approach: Structural texture analysis takes into account the hierarchy of spatial arrangements of texture primitives (macro-textures). The advantage of this approach is that it provides a good symbolic description of the image. Unlike the other three approaches, there are

not many methods that have been proposed in the literature. The one that can be considered to be powerful is the local binary pattern (LBP) [18] which has proved its efficiency in face description and recognition. Other methods which can be mentioned are the analyses of texture based on mathematical morphology [19] and on texton theory [20].

In the scope of remote sensing imagery, texture analysis plays a crucial role in several applications such as image interpretation, segmentation, classification and change detection which serve for land-cover and land-use observation, understanding and monitoring. Many methods belonging to the four mentioned approaches have been proposed to tackle this task in the literature from which the GLCM and wavelet-based techniques are the most widely used. Certain efforts have been made to extend these traditional approaches to VHR satellite images including both optical images (panchromatic, multispectral) and SAR data. For example, in [34, 35], the authors propose to retrieve the urban signatures and forest structures based on GLCM textural features extracted from VHR panchromatic images. In [36, 37], Haralick parameters are combined with color information for analyzing multispectral textures. Or in [38], GLCM features are exploited to characterize textures within SAR sea ice images. Next, the use of wavelet-based techniques has been proposed in a great amount of research work such as in [39–41]. Within the later chapters of this thesis, more related references will be mentioned according to the specific application dedicated to each chapter.

It should be noted that textural features cannot be described and extracted from a single image pixel. Indeed, a neighborhood (i.e. image patch) around each understudied pixel is taken into consideration for texture characterization. From this remark, since most of the traditional approaches exploit dense neighborhoods to estimate their parameters or exploit the whole image to perform transform techniques, they usually need to assume that the acquired image verifies the stationary or locally stationary conditions. For example, the local histogram approach and the GLCM technique require the first and second-order stationary conditions to generate their statistical textural signatures. Or, wavelet-based methods exploit all image pixels to perform their multiscale transformation. This issue may lead to the failure of these classical dense methods when applied to VHR image data which may not respect the stationarity hypothesis anymore. We will observe their poor performance during several experimental studies in the later chapters. That is the reason why we do not choose to pursue and improve any of the traditional approaches mentioned above and why we decide to propose a novel strategy relevant to the context of the thesis.

1.3 Motivation and objective

The main motivation in this thesis work is to propose a novel method for texture analysis which is relevant and adapted for VHR remote sensing images. Due to the fact the classical approaches may encounter some challenges to deal with local textures where the stationarity hypothesis is weakly verified, we would like to perform a non-dense approach based on characteristic points (i.e. keypoints, feature points) only, not on the whole image's pixels. In fact, an approach based on characteristic points does not require an assumption of stationary conditions since only the interaction and characteristics of keypoints are taken into account.

One can imagine that instead of using a dense neighborhood including all neighboring pixels around each pixel to study and describe its textural features, the expected keypoint-based approach (i.e. pointwise approach) only considers and exploits information from some feature points around the pixel. Hence, another advantage of this approach is its ability to accelerate the calculation time in order to deal with the huge volume of VHR image data.

It is worth noting that characteristic points exploited in this work must be able to provide a good representation of the image. They are required to capture most of the important information from the image content which is useful for texture characterization. In other words, they need to cover and appear within all texture zones from the image to be processed. Our work first looks for this kind of feature points and proves their adaptation and relevance for texture analysis task. A lot of keypoint extraction methods have been proposed in the literature. Do we continue to use existing keypoint types or search for other types of characteristic points? To answer this question, a detailed introduction and description of the proposed pointwise approach will be provided in the next chapter.

We now summarize the main objective and motivation of this work:

- Propose a novel approach for texture representation and analysis based on characteristic points, namely the pointwise (PW) approach.
- Study and investigate available tools (theories, techniques) which can be combined with and adapted to the proposed pointwise approach.
- Propose relevant and strategic frameworks in order to study and perform different applications of remote sensing imagery based on texture analysis including:
 - Texture-based segmentation and classification from VHR optical images (i.e. panchromatic and multispectral) acquired from forestry, agricultural, rural and peri-urban areas.
 - Texture tracking applied to the detection of land-cover changes after a disaster or in agricultural activities, as well as the detection of glacier displacements using SAR image data.
 - Texture retrieval from a large VHR image database applied to the detection of vine parcels in the scope of vineyard cultivation using optical panchromatic images.
- Evaluate the proposed frameworks and validate the results using real remote sensing data via journal publications and conference communications.
- Consider promising and perspective work based on the proposed approach for future research interventions.

1.4 Thesis overview and organization

As mentioned previously, this thesis work contributes to the field of texture analysis and characterization in the scope of image processing applied to remote sensing imagery. Several individual frameworks are proposed to tackle different texture-based applications using remote sensing images including both optical (panchromatic and multispectral) and SAR data. For the sake of clarity, we organize the thesis in four main parts.

Part I consists of two chapters from which a general introduction of the thesis is firstly described in this chapter. Then, *Chapter 2* presents the proposed pointwise approach for texture analysis which is the main contribution of this work.

The next two parts show how the proposed pointwise approach is exploited and embedded into different frameworks for texture analysis and description task applied to various applications including texture-based image retrieval, segmentation and classification, or texture tracking for change and displacement detection using remote sensed images.

Part II provides the combination of the pointwise approach with signal processing on graphs and includes three chapters:

- *Chapter 3* briefly reviews some basic notations and principles of signal processing on weighted graphs.
- *Chapter 4* describes how graph theory can be incorporated with the pointwise approach to characterize textural features and to perform texture-based classification task applied to VHR optical remote sensing images including both panchromatic and multispectral data.
- *Chapter 5* provides another combination of graph theory and the pointwise approach for texture tracking applied to the detection of land-cover changes and the detection of glacier displacements using Synthetic Aperture Radar (SAR) data.

Part III takes into consideration structural features from the image for texture analysis. By using the pointwise approach to encode radiometric and structural information, two novel local descriptors are proposed for texture characterization and discrimination, in particular to deal with oriented textures. This part consists of two chapters:

- *Chapter 6* proposes the PW-COG (i.e. pointwise covariance matrix of oriented gradients) descriptor for which we perform a pointwise approach on the construction of the feature covariance matrix. This descriptor is embedded into both unsupervised and supervised classification frameworks applied to VHR panchromatic images.
- *Chapter 7* develops the local extrema descriptor (LED). By investigating its feature space, we tackle texture retrieval task from VHR optical image databases for the application of vineyard detection in agriculture field.

Finally, **Part IV** including *Chapter 8* concludes the thesis and discusses some perspective work. We note that within this manuscript, the first page of each chapter illustrates the associated papers which are proposed to validate the chapter contribution.

Introduction to a pointwise approach for texture analysis

The world is continuous, but the mind is discrete.

David Mumford

2.1	Introduction	29
2.2	Extraction of characteristic points	29
2.3	Sensitivity to parameter	33
2.4	Why not other types of keypoints?	33
2.5	Conclusion	35

2.1 Introduction

As introduced in the first chapter, our primary proposition in this thesis work is to develop a pointwise approach for texture analysis from VHR remote sensing images. The key element of such an approach is to search for relevant characteristic points (i.e. keypoints) which have the capacity to represent and capture important properties of textural features within the image content. This chapter involves a detailed introduction of our proposed strategy. In Section 2.2, we describe the technique to extract characteristic points and prove their relevance to the thesis context. Next, we discuss the sensitivity of the approach to its parameter in Section 2.3. Section 2.4 then explains why the proposed keypoints are selected rather than other types of feature points existing in the literature. Section 2.5 concludes the chapter.

2.2 Extraction of characteristic points

Our study is inspired from the concept of the Empirical Mode Decomposition (EMD) [22–24] which is a decomposition method for non-stationary signals. In contrast to the Fourier transform and wavelet transform which break down a signal into several components based on spectral analysis, the EMD decomposes the signal directly from the time domain. It works based on producing smooth envelopes defined by local maxima and minima of a signal. Then, the so-called Intrinsic Mode Functions (IMFs) are obtained by subtracting the mean of these envelopes from the initial signal (after a number of iterations). An illustration can be found in Figure

2.1. Here, we display only the first iteration to show how the local maxima envelope E_{\max} and local minima envelope E_{\min} are produced, and how the residue is computed by subtracting the mean envelope E_M from the signal. The process continues on the residue to obtain the first IMF (with a stopping criterion) of the decomposition. More details about the EMD strategy, readers are invited to consult the related papers [22–24]. In short, the EMD generates a set of IMFs and a final residue which allow us to analyze the smooth approximation and multiscale details of the signal. This principle of EMD reveals that a non-stationary signal may be decomposed and characterized using the local maxima and minima extracted from the signal. These local extrema are capable to capture significant information of the signal content. Hence, they may be sufficient for a non-dense approach to represent and analyze non-stationary signals.

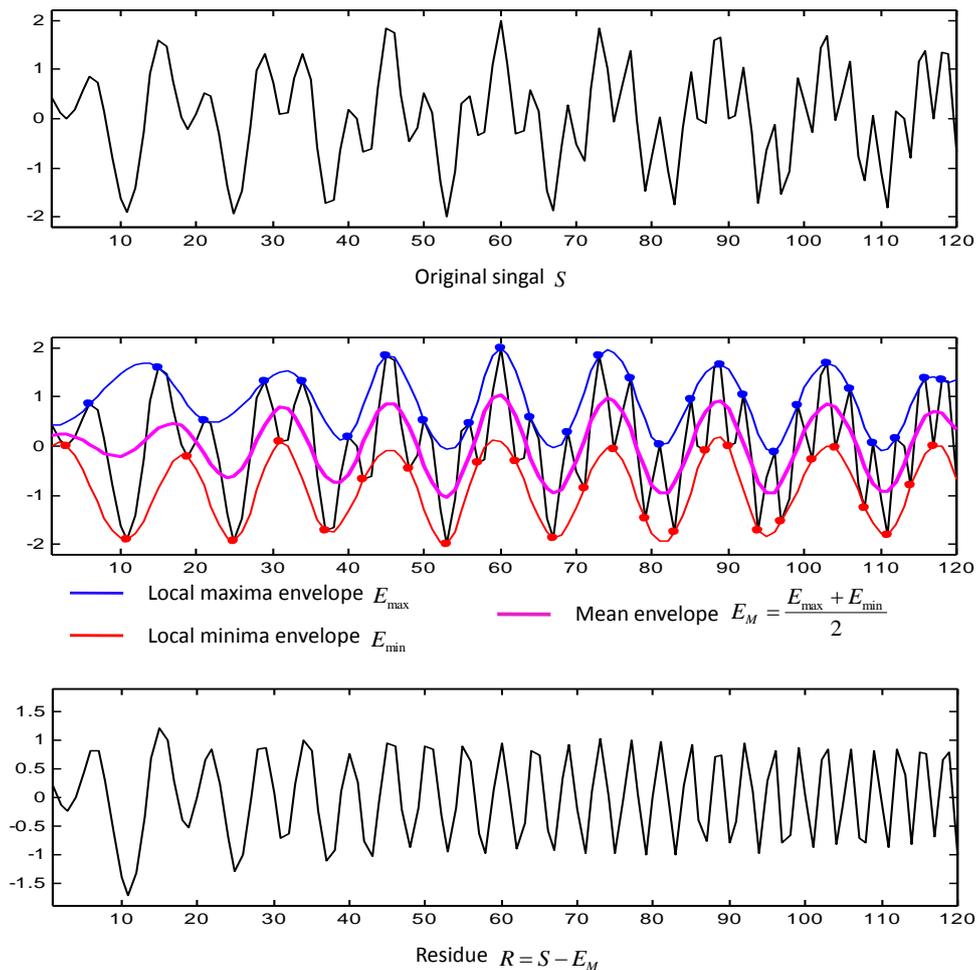


Figure 2.1 – Example of the first iteration to calculate the first IMF using the EMD method.

In this thesis work, we propose to perform a texture analysis approach based on the *local maximum* and *local minimum pixels* (in terms of intensity) extracted from the image. According to this point of view, an image texture is formed by a spatial arrangement of pixels (on the image plane) which hold some variations of intensity. Hence, different textures are reflected by different types of pixels' spatial arrangement and intensity variation. These meaningful arrangements and variations can be approximately represented and discriminated by the local maximum and local minimum pixels. To clarify this remark, let us display and analyze Figure 2.2.

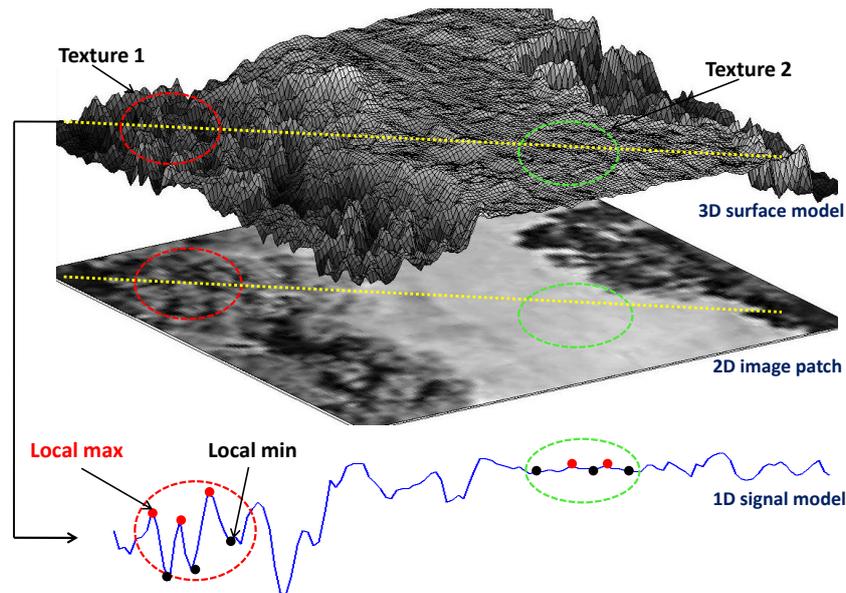


Figure 2.2 – Texture representation and discrimination using the local maximum and local minimum pixels within a pointwise approach.

The figure provides an example of different texture zones from a grayscale image crop. Let us consider two distinctive textures localized by the red and green circles. The 3-D surface model on the top is built by using the image intensity as surface height. The 1-D signal in blue at the bottom is obtained by tracing the yellow line across the image plane. From this 1-D model, we compare the first texture signal with high variation of amplitude to the second one which is smoother and less contrasted. In order to characterize and discriminate these two textures, instead of exploiting two entire signals inside the two circles, we can use only the local maxima and local minima (represented by the red points and black points, respectively) by exploiting their amplitudes and the relative distances among them. In case of 2-D images, the intensity and spatial information, represented by some measures of geometric distances and orientations, of the local maxima and minima will be taken into account to represent and characterize different textural features. This principle leads to our pointwise approach for texture analysis in this work.

We now describe the technique to extract the local maximum and local minimum pixels from a grayscale image. A pixel is supposed to be a local maximum (resp. local minimum) if it has the highest (resp. lowest) intensity within a search window centered at it. Denote $S_{\omega}^{\max}(I)$ (resp. $S_{\omega}^{\min}(I)$) the set of local maximum (resp. local minimum) pixels extracted from an image I using a search window of size $\omega \times \omega$, we define:

$$p \in S_{\omega}^{\max}(I) \Leftrightarrow \left\{ I(p) = \max_{q \in \mathcal{N}_{\omega}(p)} I(q) \right\}, \quad (2.1)$$

$$p \in S_{\omega}^{\min}(I) \Leftrightarrow \left\{ I(p) = \min_{q \in \mathcal{N}_{\omega}(p)} I(q) \right\}, \quad (2.2)$$

where $I(p)$ denotes the intensity value of pixel p and $\mathcal{N}_{\omega}(p)$ represents a set of neighboring pixels of p within the $\omega \times \omega$ search window. For implementation, the sliding window of size $\omega \times \omega$ pixels is shifted pixel by pixel on the image plane. If the centered pixel has the highest (resp. lowest) intensity, it is detected as a local maximum (resp. local minimum) point. In Figure 2.3, we show

an example of detecting the local maxima (in red, on the left) and the local minima (in blue, on the right) from an image patch using a 3×3 search window. From the figure, each square is a pixel on the image plane and the numerical value represents the pixel intensity. By applying Equations (2.1) and (2.2) with $\omega = 3$, 16 local maxima and 14 local minima are detected as observed.

15	23	12	45	81	63	21	17	20	4
42	9	25	17	42	12	28	33	56	41
12	26	91	33	24	18	8	30	12	17
66	7	71	41	33	44	52	45	47	44
8	22	32	22	81	33	45	12	31	29
56	10	55	23	13	8	15	32	65	28
5	44	57	52	61	17	56	56	23	33
16	26	42	3	15	78	22	71	21	74
81	20	19	32	41	46	45	51	49	23
14	37	12	52	74	42	23	35	33	48

15	23	12	45	81	63	21	17	20	4
42	9	25	17	42	12	28	33	56	41
12	26	91	33	24	18	8	30	12	17
66	7	71	41	33	44	52	45	47	44
8	22	32	22	81	33	45	12	31	29
56	10	55	23	13	8	15	32	65	28
5	44	57	52	61	17	56	56	23	33
16	26	42	3	15	78	22	71	21	74
81	20	19	32	41	46	45	51	49	23
14	37	12	52	74	42	23	35	33	48

Figure 2.3 – Detection of local maximum pixels (in red) and local minimum pixels (in blue) from an image patch with $\omega = 3$.



Figure 2.4 – Spatial distribution and arrangement of local maximum pixels (in red) and local minimum pixels (in blue) within different texture zones from a studied region of interest (ROI) of VHR panchromatic Pléiades image at 50cm resolution, ©CNES 2012, distribution Airbus DS/Spot Image.

The second illustration aims at demonstrating the ability of the local maximum (i.e. max) and local minimum (i.e. min) pixels to represent, characterize and discriminate textures within VHR remote sensing images. We show in Figure 2.4 their spatial arrangement and distribution within different texture zones from an image region of interest (ROI). In the figure, the local max pixels are marked in red while the local min pixels are in blue. They are extracted using Equations (2.1) and (2.2) with a 3×3 search window. From the image scene, different types of

textures are present, associated to the agricultural land occupation and exploitation in which different types of plants are grown in various orientations and various densities. As we can observe, the arrangement and alignment of the extracted local max and local min pixels allow us to approximately recognize and discriminate those textures. Therefore, we can visually perceive their capacity to represent and characterize textural features from VHR images. During the later chapters of this manuscript, the proposed pointwise approach based on these points will be integrated into different frameworks to tackle several texture-based applications. We note that the extraction of local extrema points will be adapted for each type of remote sensing data (multispectral, SAR) dedicated to each chapter.

2.3 Sensitivity to parameter

The two equations (2.1) and (2.2) point out that the number of local max and local min pixels detected from an image depends on the value of ω , the search window size. This is an advantage of the proposed pointwise approach since only one parameter is required to perform the keypoint extraction. Figure 2.5 shows an example of local max and local min keypoints detected from a forest zone (texture 1 on the top) and from an oriented planting zone (texture 2, at the bottom) using three different search window sizes of 3×3 , 7×7 and 11×11 pixels. Again, the local maxima are marked in red and the local minima are in blue from the figure. We observe that the higher ω is set, the coarser density of local max and local min keypoints is obtained, and vice versa. Moreover, there is no rule to predict the number of detected keypoints in function of ω value. It depends on the intrinsic structure of the texture itself. Nevertheless, given an image I and two window sizes ω and ω' , one can find that:

$$\omega' \geq \omega \Leftrightarrow \begin{cases} |S_{\omega'}^{\max}(I)| \leq |S_{\omega}^{\max}(I)| \\ |S_{\omega'}^{\min}(I)| \leq |S_{\omega}^{\min}(I)|, \end{cases} \quad (2.3)$$

where $|\mathcal{S}|$ represents the cardinality of the set \mathcal{S} (i.e. the number of local maximum or local minimum pixels in the dedicated set).

Another remark is that if ω is set to 1, all the pixels from the image will become local maxima as well as local minima to be detected. In this case, we get back to the classical dense approach which takes into account all image pixels from the image:

$$\omega = 1 \Leftrightarrow S_{\omega}^{\max}(I) = S_{\omega}^{\min}(I) = I. \quad (2.4)$$

Last but not least, our perception of a texture from an image always relates to the notion of scale. Within this pointwise approach, varying the window size ω provides different densities of the local maxima and local minima detected from the image. As observed in Figure 2.5, the three different densities of keypoints involve in fact the scaling issue from which the local extrema are exploited to approximately represent and characterize each texture. To this end, using different window sizes to detect local extrema keypoints may allow us to perform a multiscale pointwise analysis of texture within the spatial domain of images.

2.4 Why not other types of keypoints?

The two previous sections have described our technique to detect characteristic points which are the local maximum and local minimum pixels and proved their suitability to perform a pointwise texture analysis approach. From here, one may wonder about the possibility of using other types

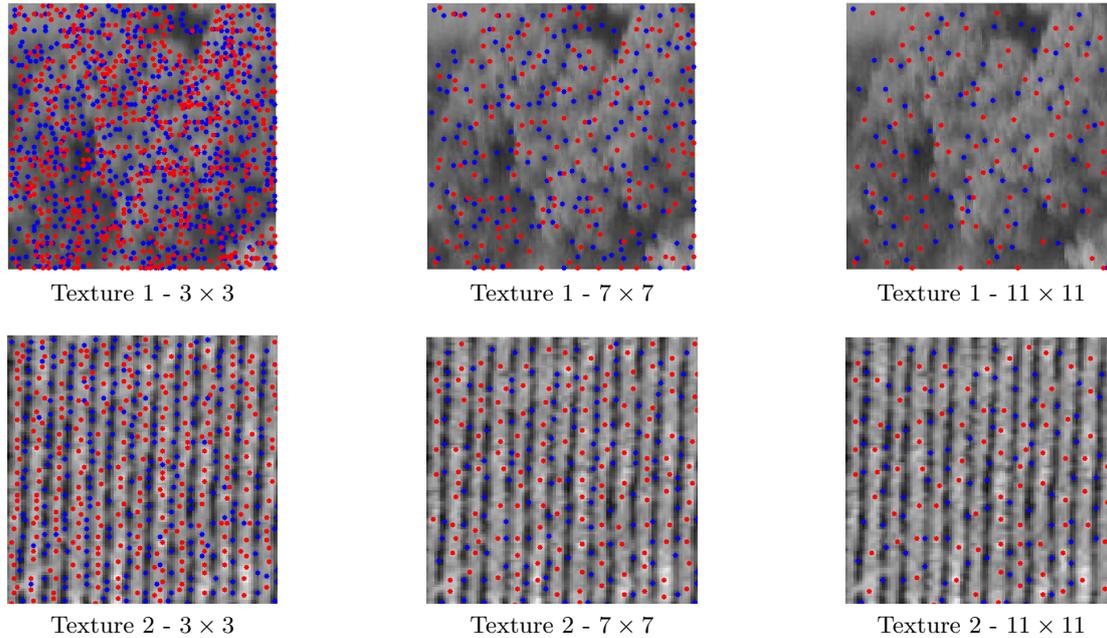


Figure 2.5 – Local max/min detection for two texture patches (140×140 pixels) with 3 different search window sizes. The local max pixels are marked in red and the local min pixels are in blue. The image patches are extracted from a VHR Pléiades image at 50cm resolution, ©CNES.

of keypoints existing in the literature such as the Harris corner points [42], the Scale-invariant feature transform (SIFT) [25], or the Speeded-up robust features (SURF) [26] (see a survey in [27]). In fact, after investigating these feature points, we found that they are not relevant to perform our pointwise approach. The reason is that in our work, characteristic points are expected to capture most of the significant information from the image content. They need to appear within all image regions where textures may exist. We have showed that the local max and local min keypoints can be detected from any variation of intensity. Hence, they are capable to cover all texture zones inside the image and become relevant for our strategy. Meanwhile, the above interest points (Harris, SIFT, SURF) are usually detected from the image features such as edges, corners and salient objects which are useful for applications like image registration and object recognition. They may not be detected from image regions where textures occur quite homogeneous and smooth. Thus, any texture analysis method based on them may encounter the problem of missing points from some textural zones. We now illustrate an example to make clear the irrelevance of those points compared to our proposed approach.

Figure 2.6 shows the distribution of three types of keypoints including the local extrema, Harris and SIFT points on the image plane. About 3000 points (marked in red) are detected in each case. From Figures 2.6(c) and 2.6(d), we observe the lack of points from the flat grass-field regions yielded by the Harris and SIFT techniques. In fact, these regions still involve a smooth texture which needs to be characterized. If the Harris or SIFT keypoints are exploited for texture analysis, these texture zones will not be taken into account. Meanwhile, the proposed local extrema keypoints (Figure 2.6(b)) are detected from all image zones to cover all types of texture. Hence, they are more relevant for the expected pointwise strategy.

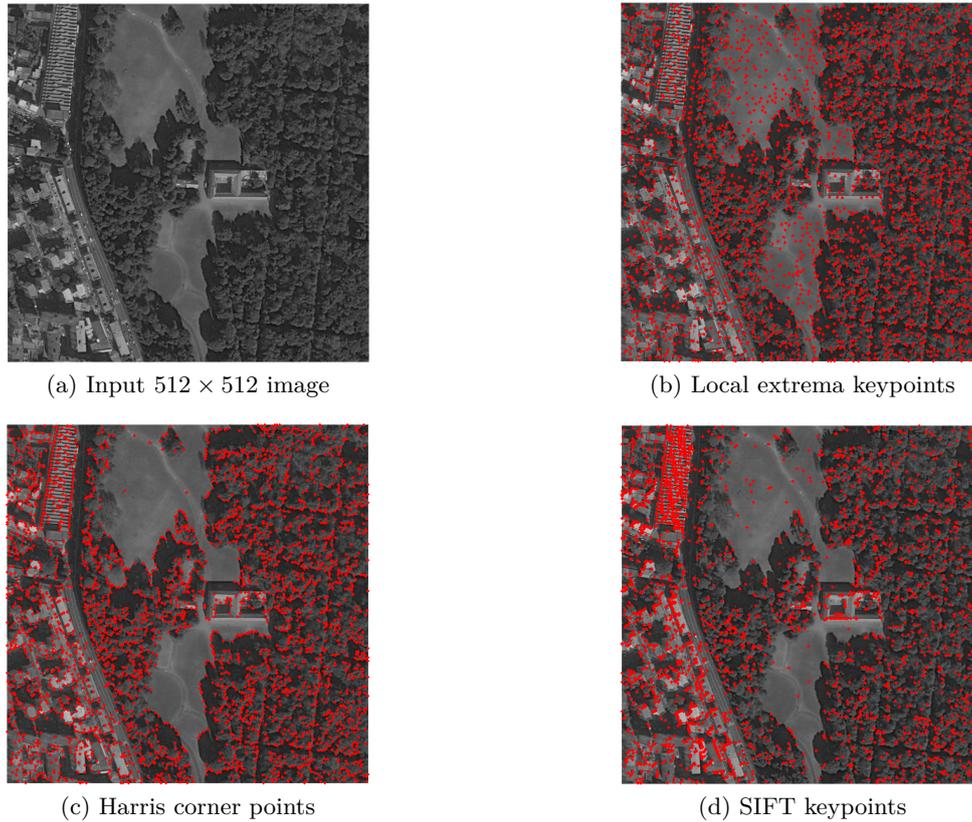


Figure 2.6 – Distribution of keypoints on the image plane: (b) local extrema points; (c) Harris corner points; (d) SIFT keypoints. In all cases, the number of keypoints is approximately 3000. The image is extracted from a VHR Pléiades image at 50cm resolution, ©CNES.

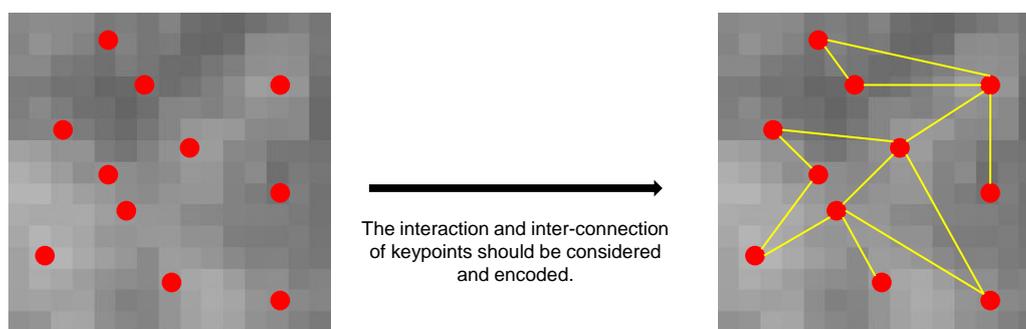
2.5 Conclusion

We have described the technique to extract the local maximum and local minimum pixels which are considered as characteristic points to perform our pointwise approach for texture analysis. These keypoints are selected thanks to their capacity of cover and appear within all texture zones from VHR images. Moreover, they are easy to be detected by sliding the search window on the image plane and comparing the pixel intensities. The fact that only one parameter (the window size ω) is involved simplifies the problem of parameter tuning for this keypoint extraction technique. Within the later chapters of the thesis (Chapters 4, 5, 6 and 7), our proposed pointwise approach will be embedded into relevant frameworks to tackle different applications of remote sensing images based on texture characterization. The effectiveness of local maximum and local minimum pixels will be confirmed and validated by each chapter's contribution.

Part II

Pointwise approach combined with graph theory

In the two previous chapters, we have presented our motivation to carry out the pointwise approach for texture analysis. The local maximum and local minimum pixels have been selected to perform such an approach. The question now is how to extract information from these characteristic points. By observing the non-dense distribution of keypoints on the image plane, the first idea coming to our mind is the fact that the interaction and inter-connection among them should be taken into account. These can be done by using the graph theory. Therefore, in this part, we propose to perform the pointwise approach encoded by weighted graph structures. Then, the concept of signal processing on graphs will be exploited for texture analysis purpose.



This part involves three up-coming chapters. We first review the principle of signal processing on graphs in Chapter 3. The idea of constructing graphs from keypoints is also presented. Next, we deliver a texture analysis framework for VHR optical remote sensing images based on the combination of pointwise approach and graph theory in Chapter 4. Chapter 5 then proposes a pointwise graph-based model to tackle texture-based land-cover change detection and glacier displacement detection tasks.

Review of signal processing on graphs

A fact acquires its true and full value only through the idea which is developed from it.

Justus Von Liebig

3.1	Introduction	39
3.2	Weighted graph and characteristics	40
3.2.1	Definition	40
3.2.2	Spectral domain of graph	41
3.2.3	Spectral graph clustering	41
3.3	Signal processing on graphs	42
3.3.1	On vertex domain	42
3.3.2	On spectral domain	43
3.4	Illustration: graph for image	44
3.4.1	Local graph for image decomposition	44
3.4.2	Non-local graph for image denoising	46
3.5	Motivation: graph for keypoints	46
3.6	Conclusion	48

3.1 Introduction

Signal processing on graphs has become an emerging field with several applications in many diverse domains including social, energy, transportation, sensor and neuronal networks [43]. One of the main contributions of our work is to incorporate the proposed pointwise approach with graph theory in order to exploit some available tools of signal processing on graphs for the applications of texture-based image classification and change detection, which will be detailed later in Chapter 4 and Chapter 5. In this chapter, some basic notions and characteristics of weighted graph (i.e. similarity graph) are first provided. Next, some principles of signal analysis and processing on both vertex and spectral domains of graph are briefly reviewed in Section 3.3. Then, Section 3.4 illustrates the application of spectral graph theory for image multiscale decomposition and image denoising before Section 3.5 provides our motivation of using weighted graph to encode characteristic points extracted from the image. Finally, Section 3.6 concludes the chapter with some discussions and addresses preliminary instructions of how graph signals will be exploited within the next two chapters.

3.2 Weighted graph and characteristics

3.2.1 Definition

A weighted graph $\mathcal{G} = \{V, E, w\}$ consists of a set of vertices $V = \{v_i; i = 1, \dots, N\}$ connected by a set of edges E and their associated weights w involving a measure of similarity between vertices. The adjacency matrix \mathcal{W} (i.e. similarity matrix, matrix of weights) of \mathcal{G} is defined:

$$\mathcal{W}_{ij} = \begin{cases} w(i, j) & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases} \quad (3.1)$$

In the scope of our work, only undirected and unlooped graphs are considered so that $w(i, j) = w(j, i)$ and $w(i, i) = 0, \forall i, j \in \{1, \dots, N\}$. Figure 3.1 shows an example of a weighted graph \mathcal{G} with $N = 4$ vertices, $V = \{v_1, v_2, v_3, v_4\}$ and $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_3, v_4)\}$.

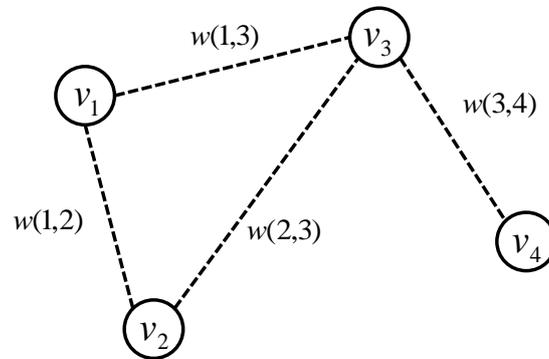


Figure 3.1 – Example of a weighted, undirected and unlooped graph.

Some basic matrices in graph theory are defined:

- The degree matrix \mathcal{D} which is diagonal:

$$\begin{cases} \mathcal{D}_{ii} = \sum_j \mathcal{W}_{ij}, \\ \mathcal{D}_{ij} = 0, \forall i \neq j. \end{cases} \quad (3.2)$$

- The Laplacian matrix:

$$\mathcal{L} = \mathcal{D} - \mathcal{W}. \quad (3.3)$$

- The normalized Laplacian matrix:

$$\mathcal{L}_{\text{nor}} = \mathcal{D}^{-\frac{1}{2}} \mathcal{L} \mathcal{D}^{-\frac{1}{2}} = \mathcal{I} - \mathcal{D}^{-\frac{1}{2}} \mathcal{W} \mathcal{D}^{-\frac{1}{2}}, \quad (3.4)$$

where \mathcal{I} is the $N \times N$ identity matrix.

- The random walk matrix:

$$\mathcal{P} = \mathcal{D}^{-1} \mathcal{W}. \quad (3.5)$$

3.2.2 Spectral domain of graph

Spectral graph analysis is usually performed based on the eigen decomposition of the Laplacian matrix \mathcal{L} [44]. Some studies also exploit the adjacency matrix \mathcal{W} , the random walk matrix \mathcal{P} , or the normalized Laplacian matrix \mathcal{L}_{nor} for the eigen decomposition to perform spectral analysis on graphs. For more details about their differences, readers are invited to see papers [44,45]. In our study, we focus on the use of the unnormalized version of Laplacian matrix \mathcal{L} for this task, which is originally proposed by F.R.K. Chung in [44].

The eigen decomposition of \mathcal{L} is given by:

$$\mathcal{L}\chi_k = \lambda_k\chi_k, \quad k = 0, \dots, N - 1. \quad (3.6)$$

Since \mathcal{L} is symmetric, positive and semi-definite, the following properties are considered:

- $\{\lambda_k\}_{k=0\dots N-1}$ is a non-negative eigenvalue set where:

$$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{N-1}.$$

- $\{\chi_k\}_{k=0\dots N-1}$ forms an orthogonal eigenvector basis.
- $\sigma(L) = \{\lambda_0, \lambda_1, \dots, \lambda_{N-1}\}$ refers to the entire graph spectrum [44] in which a small eigenvalue λ_k represents a low frequency in graph spectral domain, and vice versa.

Notion of frequency: In graph spectral domain, the Laplacian eigenvalues and eigenvectors provide a similar notion to the classical Fourier frequency. In details:

- + With $\lambda_0 = 0$, the associated eigenvector χ_0 is a constant vector equivalent to the DC component across the graph.
- + For a low frequency λ_k close to 0, the values of the corresponding eigenvector χ_k vary slowly and smoothly across the graph. If two vertices v_i and v_j are connected with an important weight $w(i, j)$, corresponding values at their locations, i.e. $\chi_k(i)$ and $\chi_k(j)$, are likely to be similar.
- + For a high frequency λ_k far from 0, corresponding eigenvector oscillates more rapidly. If two vertices v_i and v_j are connected with a large weight $w(i, j)$, $\chi_k(i)$ and $\chi_k(j)$ are likely to have dissimilar values.

3.2.3 Spectral graph clustering

The spectral clustering algorithm [45, 46] separates graph vertices into different groups based on their pairwise similarity and on the aforementioned notion of frequency in graph spectral domain. Given a set of N data points x_1, \dots, x_N and a distance measure metric, following steps are activated for spectral clustering algorithm dividing the data into C clusters:

- Construct a weighted graph \mathcal{G} connecting N given points where each data point x_i corresponds to a graph vertex v_i and the adjacency matrix $\mathcal{W} \in \mathbb{R}^{N \times N}$ is calculated based on the similarity between data points using the given distance measure.
- Compute the graph Laplacian matrix \mathcal{L} .
- Compute the first C eigenvectors $\{\chi_k\}_{k=0, \dots, C-1}$ corresponding to the C smallest eigenvalues $\{\lambda_k\}_{k=0, \dots, C-1}$ via the eigen decomposition of \mathcal{L} .

- Consider the matrix $U = [\chi_0, \dots, \chi_{C-1}] \in \mathbb{R}^{N \times C}$ as feature space of N data points where each point associates with a feature vector in \mathbb{R}^C .
- Cluster the above N points into C clusters by performing the *K-means clustering* algorithm [47] on their feature vectors.

The above algorithm acts on the unnormalized graph Laplacian matrix \mathcal{L} so that it is called the unnormalized spectral clustering. As mentioned previously, another version called normalized spectral clustering can be found in the literature when \mathcal{L} is substituted by \mathcal{L}_{nor} . We remind that in the scope of our work, only the unnormalized spectral clustering using \mathcal{L} is considered.

3.3 Signal processing on graphs

Let $f = \{f(n) \in \mathbb{R}; n = 1, \dots, N\}$ be the signal or function on graph vertices called graph signal (see Figure 3.2). The analysis and processing of f on graph can be performed on both the vertex and spectral domains. In this section, some important characteristics and transforms which are exploited in the scope of the thesis are reviewed. For further details about vertex and frequency analysis of signal on graphs, readers are invited to consult papers [43, 48, 49].

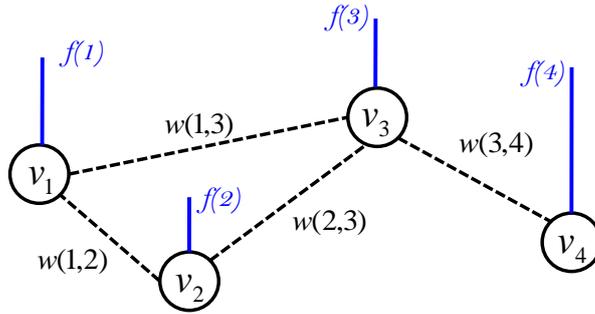


Figure 3.2 – A signal f on graph \mathcal{G} . Each vertex v_i encapsulates a signal sample $f(i)$.

3.3.1 On vertex domain

The analysis and processing of signal on graph vertex domain are mainly performed with the help of the defined matrices in Section 3.2.1. Given a signal on graph $f \in \mathbb{R}^N$, the following equations are obtained:

$$(\mathcal{W}f)(n) = \sum_{v_m \sim v_n} w(n, m) f(m), \quad (3.7)$$

$$(\mathcal{P}f)(n) = \frac{1}{\sum_{v_m \sim v_n} w(n, m)} \sum_{v_m \sim v_n} w(n, m) f(m), \quad (3.8)$$

$$(\mathcal{L}f)(n) = \sum_{v_m \sim v_n} w(n, m) [f(n) - f(m)], \quad (3.9)$$

$$f^T \mathcal{L} f = \sum_{v_m \sim v_n} w(n, m) [f(n) - f(m)]^2, \quad (3.10)$$

where the notation $v_m \sim v_n$ means that two vertices v_m and v_n are connected (i.e. $(v_m, v_n) \in E$).

It should be noted that within a weighted graph, a vertex diffuses its information to its neighbors and, at the same time, it receives the information diffused from other vertices, according to their weight functions. This property makes the similarity graph model become a very effective tool not only for representing information from a signal but also for understanding its intrinsic inter-connection structure. Therefore, regarding to the above equations, following remarks can be made:

- $(\mathcal{W}f)(n)$ and $(\mathcal{P}f)(n)$ perform the concentration of information from neighboring vertices to the understudied vertex v_n . As a result, we can consider $\mathcal{W}f$ as a filtering operator (low-pass) performed in graph vertex domain and $\mathcal{P}f$ as a normalized version of this filter.
- $\mathcal{L}f$ can be considered as a difference operator.
- $f^T \mathcal{L}f$ represents the total variation of the signal f on graph.

The above formulations help to analyze the signal f based on the intrinsic structure coming from graph vertex connections given by their weights.

3.3.2 On spectral domain

The analysis of signal on graphs can be also performed on the frequency (spectral) domain using the eigen decomposition of the Laplacian matrix. Instead of directly exploiting the eigenvector set of the graph Laplacian matrix as in the case of spectral clustering (Section 3.2.3), researchers have triggered some transformation tools on the graph spectral domain analogous to the classical Fourier frequency domain of signals.

First and foremost, the spectral graph Fourier transform (SGFT) is introduced. Consider a signal $f \in \mathbb{R}^N$, we remind the classical discrete Fourier transform (DFT):

$$\hat{f}(k) = \sum_{n=1}^N f(n)e^{-2j\pi nk/N}. \quad (3.11)$$

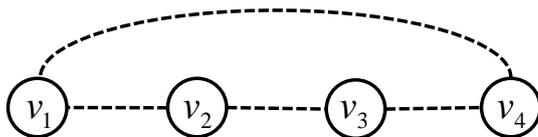


Figure 3.3 – A ring graph.

Now, let us consider a ring graph \mathcal{G}_r as in Figure 3.3 (here $N = 4$ vertices). If we perform the eigen decomposition of the normalized Laplacian matrix of \mathcal{G}_r as in Equation (3.6), we obtain an orthogonal eigenvector basis χ_k which is exactly the same as the Fourier basis:

$$\forall k \in \{0, 1, \dots, N-1\}, \quad \chi_k = \left[1 \quad \omega_N^k \quad \omega_N^{2k} \quad \dots \quad \omega_N^{(N-1)k} \right]^T, \quad \omega_N = e^{j2\pi/N}. \quad (3.12)$$

Hence, there exists an explicit link between the spectral analysis of discrete signal and signal processing on ring graphs. The analysis of a signal on a ring graph in the spectral domain allows us to perform the classical DFT of the signal. If we investigate any other graph structures, the spectral notion is extended by using other orthogonal eigenvector bases. By replacing the

exponential eigenfunction from the DFT by graph Laplacian eigenvector χ_k , the SGFT is defined as the projection of the signal f onto the eigenvector space:

$$\widehat{f}(k) = \langle f, \chi_k \rangle = \sum_{n=1}^N f(n) \chi_k^*(n). \quad (3.13)$$

For an analogue, the inverse graph Fourier transform can be generated:

$$f(n) = \sum_{k=0}^{N-1} \widehat{f}(k) \chi_k(n). \quad (3.14)$$

Another emblematic transformation introduced in [50] is the spectral graph wavelet transform (SGWT). In fact, the value of $\widehat{f}(k)$ indicates how much each eigenvector χ_k is present in the signal f . Hence, from the reconstruction of signal by the inverse SGFT in (3.14), it is possible to modify the contribution of each eigenvector by weighting \widehat{f} with the support of a filter response in graph spectral domain. Then, by changing the filter stretching level using a dilation factor t called scale, SGWT enables us to perform a multiscale analysis of signal on graph.

Thus, using a function $\widehat{g} : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ representing the transfer function of a band-pass filter in the spectral domain of graph, the authors in [50] generated the graph wavelet coefficients of a signal $f \in \mathbb{R}^N$ at vertex v_n and scale t as follows:

$$W_f(t, n) = \sum_{k=0}^{N-1} \widehat{g}(tk) \widehat{f}(k) \chi_k(n). \quad (3.15)$$

Similarly, to be analogous to classical wavelet transform, SGWT scaling function coefficients can also be generated with the help of the transfer function of a low-pass filter $\widehat{h} : \mathbb{R} \rightarrow \mathbb{R}^+$:

$$S_f(n) = \sum_{k=0}^{N-1} \widehat{h}(k) \widehat{f}(k) \chi_k(n). \quad (3.16)$$

Therefore, given a signal on vertices $f \in \mathbb{R}^N$ and a set of scales $\{s_j\}_{j=1, \dots, J}$, SGWT generates a set of graph scaling function coefficients $S_f \in \mathbb{R}^N$ and J sets of graph wavelet coefficients $W_f(t_j) \in \mathbb{R}^N$ at scale t_j . Here is the outline of SGWT frame:

$$\begin{aligned} \text{SGWT: } \mathbb{R}^N &\longmapsto \mathbb{R}^{N(J+1)} \\ f &\longrightarrow S_f + \{W_f(t_j)\}_{t_j=1, \dots, J} \end{aligned} \quad (3.17)$$

3.4 Illustration: graph for image

There are several techniques using graphs for image representation and analysis [51, 52]. In the literature, a common way to encode an image by a similarity graph is to consider each pixel as a graph vertex. We illustrate here the construction of graph and the application of spectral graph wavelet transform (SGWT) for image multiscale analysis and image denoising problems.

3.4.1 Local graph for image decomposition

Figure 3.4 shows an example of multiscale decomposition for a crop of a VHR image by graph wavelets performing on a local graph (i.e. 4-connected graph) constructed by setting each pixel as a graph vertex. Each vertex is connected to its 4 closest neighbors on the image plane with

an equal weight to 1. The graph structure can be observed on the top right of the figure. The SGWT is performed by considering the image intensity as the function on graph f . Scaling function coefficients S_f and wavelet coefficients $W_f(t_1)$ and $W_f(t_2)$ at 2 different scales t_1 and t_2 are generated according to Equations (3.16) and (3.15), respectively. We also display in Figure 3.5 the corresponding low-pass filter h and band-pass filters $g(t_1)$ and $g(t_2)$ within the graph spectral domain which are used to generate those scaling function and wavelet coefficients. At the bottom of Figure 3.4, the scaling function image (i.e. low-pass component) and the 2 wavelet images (i.e. band-pass components) can be observed. Analogous to the classical discrete wavelet transform, the low-pass component captures the coarse form and content of the image while the 2 band-pass components encapsulate fine details such as edges, contours, noise, etc.

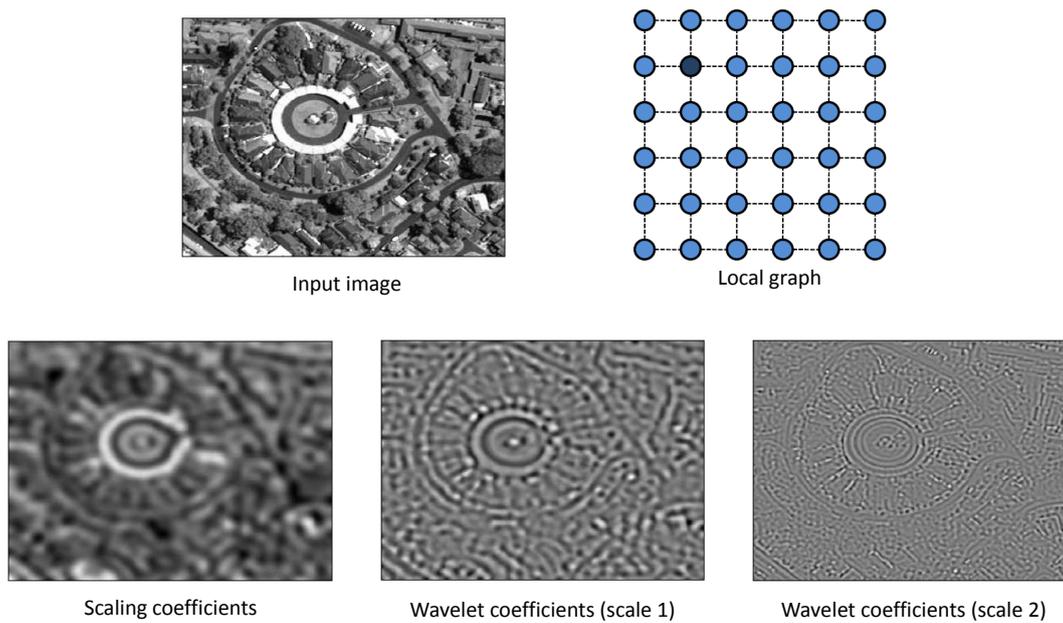


Figure 3.4 – Example of multi-scale image decomposition using wavelet transform on a local graph.

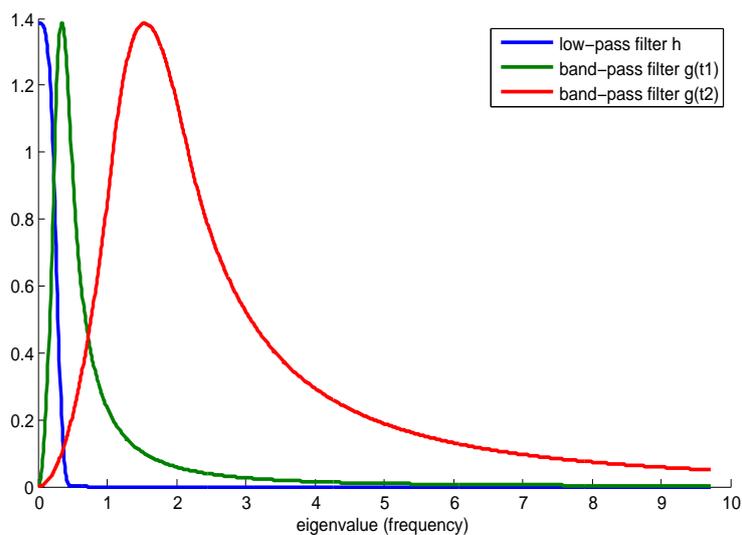


Figure 3.5 – Low-pass and band-pass filters for SGWT image decomposition in Figure 3.4.

3.4.2 Non-local graph for image denoising

Another graph structure that can be investigated for image analysis is the non-local graph. Indeed, each pixel is considered as a vertex and will be connected to other vertices according to their similarities in terms of intensity. Hence, edge weights are calculated based on the similarity between pixel intensities and can be performed by an exponential form $e^{-\|I(i)-I(j)\|}$, where $I(i)$ and $I(j)$ denote the intensities of vertices v_i and v_j . An example can be found in Figure 3.6. The non-local graph (bottom left) is constructed from the noisy image (top right). Then, the SGWT is performed on this graph. Some wavelet coefficients close to zero are thresholded to exclude noise from the image. The image is then reconstructed and shown at the bottom right of the figure. We do not bring many details or the evaluation of denoising performance since this is just an illustration for this graph chapter.

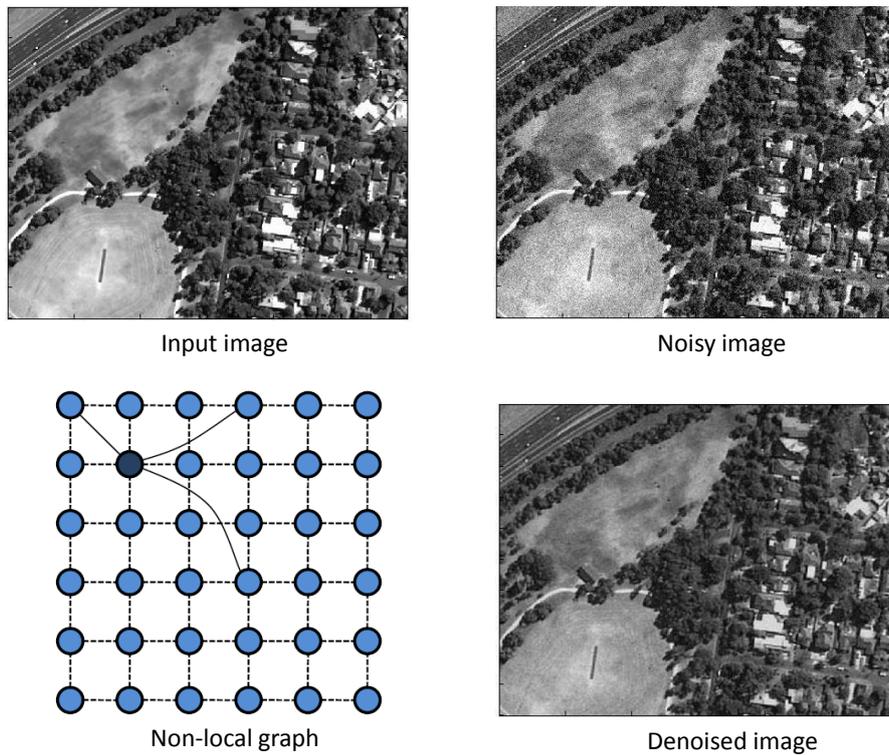


Figure 3.6 – Example of image denoising using wavelet transform on a non-local graph.

3.5 Motivation: graph for keypoints

Our main motivation in Part II is to incorporate the proposed pointwise approach (Chapter 2) into the graph theory. To do that, one of the most significant stages is to construct a weighted graph to encode a keypoint set extracted from the image. Unlike traditional methods considering each pixel as a graph vertex which have been previously illustrated, our proposition is to take into account only keypoints for graph construction.

Given a set of keypoints $S = \{p_n; n = 1, \dots, N\}$, a weighted graph $\mathcal{G} = \{V, E, w\}$ can be constructed to connect all the keypoints in S as follows:

- Vertex set: $V = S$. Each keypoint is considered to be a graph vertex. Hence, the dimension of graph \mathcal{G} is $|V| = N$.

- Edge set: there are two common ways to define the edge set E :

+ Two vertices are connected if their distance is smaller than a defined threshold κ :

$$E = \{(p_n, p_k); p_n, p_k \in V, \text{dist}(p_n, p_k) < \kappa\},$$

where $\text{dist}(p_n, p_k)$ represents a spatial distance between vertices p_n and p_k , or the distance between two feature vectors describing p_n and p_k . The latter is generally used in graph-based semi-supervised learning methods [43] and will be exploited within the scope of this thesis.

+ Each vertex is connected to its K -nearest neighbors in terms of spatial or feature-based distance:

$$E = \{(p_n, p_k); p_n, p_k \in V, p_n \in \mathcal{N}_K(p_k) \text{ and } p_k \in \mathcal{N}_K(p_n)\},$$

where $\mathcal{N}_K(p_n)$ (resp. $\mathcal{N}_K(p_k)$) denotes the set of K nearest keypoints of p_n (resp. p_k) in terms of spatial or feature-based distance.

- Weight function $w(p_n, p_k)$: involves the similarity between two vertices p_n and p_k and can be generated by the exponential function as follows:

$$w(p_n, p_k) = e^{-\gamma \text{dist}^2(p_n, p_k)}, \quad \forall (p_n, p_k) \in E.$$

where γ is a free parameter which is usually fixed to 1 within our implementation.

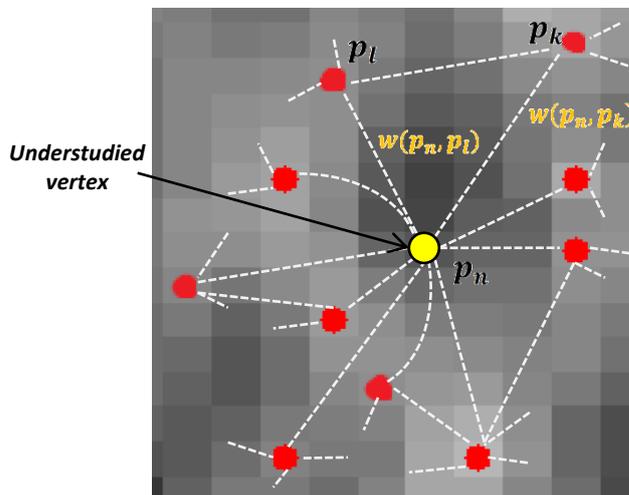


Figure 3.7 – Graph construction for keypoints.

Figure 3.7 shows an example of graph construction for keypoints on the image plane. Each keypoint p_n (yellow) is considered as a vertex and connected to its neighboring keypoints (red) by the graph \mathcal{G} . Each connection is associated to a weight representing the similarity between 2 corresponding vertices.

3.6 Conclusion

Graph structure is a useful tool to represent and manipulate different types of data including images. The vertex domain allows us to encode the interaction and inter-connection among vertices and to study how signals are distributed on and captured by them. Hence, some filtering and diffusion processes can be activated to study the coherence of information encapsulated by graph vertices, which is promising for the detection and tracking tasks. On the other hand, the spectral domain provides an analogous notion to the classical Fourier frequency domain. It enables us to perform some transformations that study the multiscale property of signals on graph, which is relevant for texture description in the scope of image processing. To this end, Table 3.1 summarizes the intervention of graph signal processing in the next two chapters which are the main contributions of Part II.

Table 3.1 – Intervention of graph signal processing in Chapter 4 and Chapter 5.

Chapter	Graph signal processing	Image data	Application
4	on spectral domain	optical (panchromatic, multispectral)	texture description, texture-based classification
5	on vertex domain	SAR (Synthetic Aperture Radar)	texture-based change detection, glacier displacement tracking

Pointwise graph-based texture description for VHR optical image classification

What you do today can improve all your tomorrows.

Ralph Marston

4.1	Introduction	50
4.2	Texture description using a pointwise approach	51
4.3	Proposed classification framework	53
4.3.1	Extraction of characteristic points for image representation and description	55
4.3.2	Weighted graph construction	56
4.3.3	Textural feature extraction	57
4.3.4	Clustering algorithm for unsupervised classification	59
4.4	Experimental results	59
4.4.1	Datasets	59
4.4.2	Experimental setup	59
4.4.3	Classification results	61
4.4.4	Performance in time consumption	65
4.4.5	Sensitivity of the proposed method	66
4.5	Conclusion	67

The content of this chapter is mainly based on the following published papers:

- M. T. Pham, G. Mercier, J. Michel, *Pointwise graph-based local texture characterization for very high resolution multispectral image classification*, IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens., vol. 8, issue 5, pp. 1962-1973, 2015.
- M. T. Pham, G. Mercier, J. Michel, *Textural features from wavelets on graphs for very high resolution panchromatic Pleiades image classification*, Revue française de photogrammétrie et télédétection, no. 208, pp. 131-136, 2014.
- M. T. Pham, G. Mercier, J. Michel, *Wavelets on graphs for very high resolution multispectral image segmentation*, Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Québec city, Canada, 2014, pp. 2273-2276.

4.1 Introduction

In the field of remote sensing imagery, texture description plays a significant role in several applications such as image interpretation, segmentation, classification or change detection serving for urban and forestry land-cover observation, monitoring and mapping [53, 54]. The aim of the present chapter is to develop a novel strategy for the analysis and characterization of textures in VHR optical satellite images to perform classification task.

As previously discussed in Chapters 1 and 2, with the emergence of VHR remote sensing technology, in both spatial and spectral terms, texture characterization task becomes more challenging due to the appearance of heterogeneous texture zones and the weak verification of stationarity hypothesis within VHR images. Classical methods such as the Local Histogram [4], the Gray Level Cooccurrence Matrix (GLCM) [6], the Gabor filter [12, 13] and the wavelet-based approaches [14, 15] are no longer relevant if their original version is applied. In fact, recent studies believe that textures are expected to be characterized and exploited under the spatial-spectral constraint in VHR remote sensing field. This issue has been taken into consideration and addressed in some research work dedicated to texture-based applications. As a result, segmentation and classification strategies based on spectral-texture combination for remote sensed data have been proposed in [55–60]. For example in [56], a scale-space representation of GLCM is performed to extract multiscale textural features. In [58], the Gabor filter banks are performed over individual bands and over complex bands of hyperspectral images to extract textural features for classification task. Or in [59], an automatic selection of Gabor filter scales combined with the spectral histogram is proposed to perform texture-based segmentation. In general, these methods are still limited to the extraction of classical textural features (e.g. Haralick features, Gabor or wavelet coefficients, etc.) for which a dense approach needs to be considered and the stationary or local stationary conditions are required.

In Chapter 2, the pointwise approach based on characteristic points which are the local maximum and local minimum pixels was proved to be relevant for texture analysis from VHR images. Such an approach does not require any stationary condition and is able to deal with a great amount of image data since we do not need to consider all the image pixels. Then, Chapter 3 showed that graph model is a useful tool to encode the interaction and inter-connection of keypoints extracted from an image. Moreover, the graph spectral domain can provide some transformation tools which are appropriate for multiscale analysis of signal. Hence, in this chapter, our motivation is to embed the pointwise approach into a graph model. Here, a weighted graph is constructed to connect keypoints based on their similarity in terms of local textures. Then, spectral characteristics of this graph will be extracted and exploited as the final textural features based on them classification stage can be performed. We investigate two outstanding spectral-graph tools for this task: the spectral graph clustering (SGC) [46] and the spectral graph wavelet transform (SGWT) [50].

The remainder of this chapter is organized as follows. Section 4.2 presents the generation of textural feature descriptor using a pointwise approach based on the local max and local min pixels. We describe in details the proposed texture-based classification algorithm using the pointwise graph-based model applied to both panchromatic and multispectral images in Section 4.3. Then, Section 4.4 provides our experimental study on VHR multispectral image data. Results yielded by the proposed framework are evaluated and compared to several classical methods in terms of classification accuracy and computational time consumption. We also investigate the sensitivity of the proposed method to its parameters at the end of the section. Finally, Section 4.5 concludes and summarizes the contribution of this chapter.

4.2 Texture description using a pointwise approach

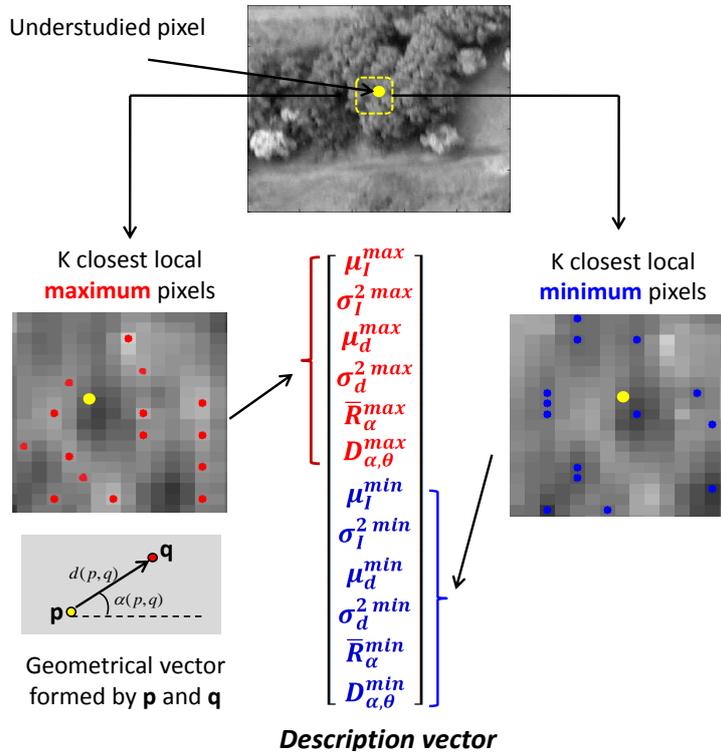


Figure 4.1 – Generation of the proposed pointwise descriptor for an understudied pixel.

In this section, we develop the pointwise (PW) feature descriptor using a non-dense approach based on the local extrema points. For a reminder, instead of using a dense neighborhood centered at each understudied pixel (i.e. keypoint), only the local maximum and local minimum points around it are taken into consideration. To characterize textural features, both intensity and geometric information of these extrema points will be encoded. First, two strategies can be considered for the research of the closest local maxima and closest local minima around each studied keypoint:

1. Fix the number of K closest local maxima and K closest local minima for each one.
2. Or, fix a window size $W \times W$ around each keypoint, then all local maxima and local minima inside that window are considered.

The first strategy takes into account the local properties around keypoints since the implicit neighborhood size considered for each one varies depending on the density of local extrema around it (i.e. the search process goes further to look for enough extrema when their density is sparse). On the other hand, by fixing the window size, the second approach considers equivalent contributions of neighboring environments for all keypoints and also better deals with outlier points. However, our experimentation shows that both strategies can provide similar performance for the studied data sets. In this chapter, we describe the first approach considering a fixed number of K closest local maxima and K closest local minima. Nevertheless, it is worth noting that a similar principle can be applied to build the proposed descriptor using the second approach.

We present now the computation of the proposed pointwise descriptor (i.e. sparse descriptor) applied to a panchromatic image. This process is outlined in Figure 4.1. The extended version for multispectral images will be addressed later. Let us consider an understudied pixel p (yellow point) located at position (x_p, y_p) on the image I and having its intensity value $I(p)$. We search for two sets of K closest local maxima and K closest local minima of p which are denoted by $S_K^{\max}(p)$, $S_K^{\min}(p)$, respectively (these local extrema are marked in red and blue from the figure). For each point q from these two sets, the pair (p, q) forms a geometrical vector having length $d(p, q)$ and angle $\alpha(p, q)$ as shown at the bottom left of the figure. The following features are extracted from the set $S_K^{\max}(p)$, noting that a similar process is applied to $S_K^{\min}(p)$:

- Mean and variance of intensities:

$$\mu_I^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} I(q), \quad (4.1)$$

$$\sigma_I^2{}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (I(q) - \mu_I^{\max}(p))^2. \quad (4.2)$$

- Mean and variance of spatial distances:

$$\mu_d^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} d(p, q), \quad (4.3)$$

$$\sigma_d^2{}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (d(p, q) - \mu_d^{\max}(p))^2, \quad (4.4)$$

where $d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$.

- Measures of the θ -dispersion and the concentration of directions which come from the theory of Directional Statistics in [29]:

$$D_{\alpha, \theta}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \{1 - \cos(\alpha(p, q) - \theta)\}, \quad (4.5)$$

$$\bar{R}_\alpha^{\max}(p) = \sqrt{\bar{C}_\alpha^2(p) + \bar{S}_\alpha^2(p)}, \quad (4.6)$$

where

$$\bar{C}_\alpha(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \cos \alpha(p, q),$$

$$\bar{S}_\alpha(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \sin \alpha(p, q),$$

$$\alpha(p, q) = \arctan \left(\frac{y_q - y_p}{x_q - x_p} \right), \quad \alpha(p, q) \in [-\pi, \pi].$$

In [29], the authors prove that the measure of concentration \bar{R}_α indicates if all vector directions are tightly clustered (\bar{R}_α close to 1) or widely dispersed (\bar{R}_α close to 0). Also, the circular variance can be defined as $1 - \bar{R}_\alpha$. Since \bar{R}_α is invariant to rotation, the second measure named directional dispersion from a given angle θ , denoted by $D_{\alpha, \theta}$, is required to distinguish the principal direction focused by different vector groups. A relationship that can be found here is $D_{\alpha, \bar{\alpha}} = \min_{\theta} \{D_{\alpha, \theta}\} = 1 - \bar{R}_\alpha$ where $\bar{\alpha}$ is the mean angle of the set $\{\alpha(p, q); q \in S_K^{\max}(p)\}$, defined in [29]. In our experiments, we set the value $\theta = 0^\circ$.

Similar features are also computed from the closest local minima set $S_K^{\min}(p)$. Now, let $\delta^{\text{PW}}(p)$ be the pointwise descriptor (i.e. pointwise signature vector) of the studied pixel p . For short, we obtain:

$$\delta^{\text{PW}}(p) = \left[\mu_I^{\max}(p), \sigma_I^2{}^{\max}(p), \mu_d^{\max}(p), \sigma_d^2{}^{\max}(p), \bar{R}_\alpha^{\max}(p), D_{\alpha,\theta}^{\max}(p), \right. \\ \left. \mu_I^{\min}(p), \sigma_I^2{}^{\min}(p), \mu_d^{\min}(p), \sigma_d^2{}^{\min}(p), \bar{R}_\alpha^{\min}(p), D_{\alpha,\theta}^{\min}(p) \right]. \quad (4.7)$$

$\delta^{\text{PW}}(p) \in \mathbb{R}^{12}$ provides a local non-dense description of textural features. It enables us to understand how the local maximum and local minimum pixels are distributed and arranged within the environment around the pixel p . There are two key parameters which should be taken into account from the computation of description vectors. The first one is the search window size ω used for detecting local max and local min points. The impact of this parameter which relates to the scale-dependent property of texture has been already discussed in Section 2.3 (Chapter 2). The second one which appears directly in all the above equations is the number K of closest extrema that are considered to describe each understudied pixel. This parameter associates with the notion of texture localization from the image. We set K from 10 to 25 in our implementation. Some experimental results will be provided in Section 4.4.5 to study the sensitivity of the algorithm to these parameters.

Similar to other feature description techniques, the proposed approach is originally applied to panchromatic images. In order to extend it to data such as multispectral or hyperspectral images, it always refers to an opened issue depending on how users integrate spectral information to form their descriptors. Our own method for multispectral data will be presented in the next section when these pointwise descriptors are integrated into a graph-based model to perform classification task.

4.3 Proposed classification framework

This section presents the complete framework for texture characterization of VHR optical images applied to classification task. In general, the proposed algorithm consists of four main stages:

1. Extraction of characteristic pixels (i.e. keypoints) for image representation and extraction of the local maxima and local minima sets for pointwise feature description.
2. Construction of weighted graph to connect keypoints using the extracted pointwise descriptors.
3. Extraction of spectral graph characteristics and transformations to obtain the final textural features.
4. Unsupervised clustering based on the extracted textural features to yield classification result.

An outline of the proposed framework is highlighted in Figure 4.2. Here, the input can be a VHR panchromatic or multispectral image, which respectively appears in the left-side or right-side first blocks (blue dashed box) of the figure. In fact, the original algorithm is developed on panchromatic images. Then, by incorporating some spectral information, it can be applied to multispectral images. We now briefly describe the framework before addressing more clearly each stage in the following subsections.

For a panchromatic input image I , the extraction of characteristic points to represent the image and the local extrema pixels to calculate pointwise texture descriptors at each keypoint can

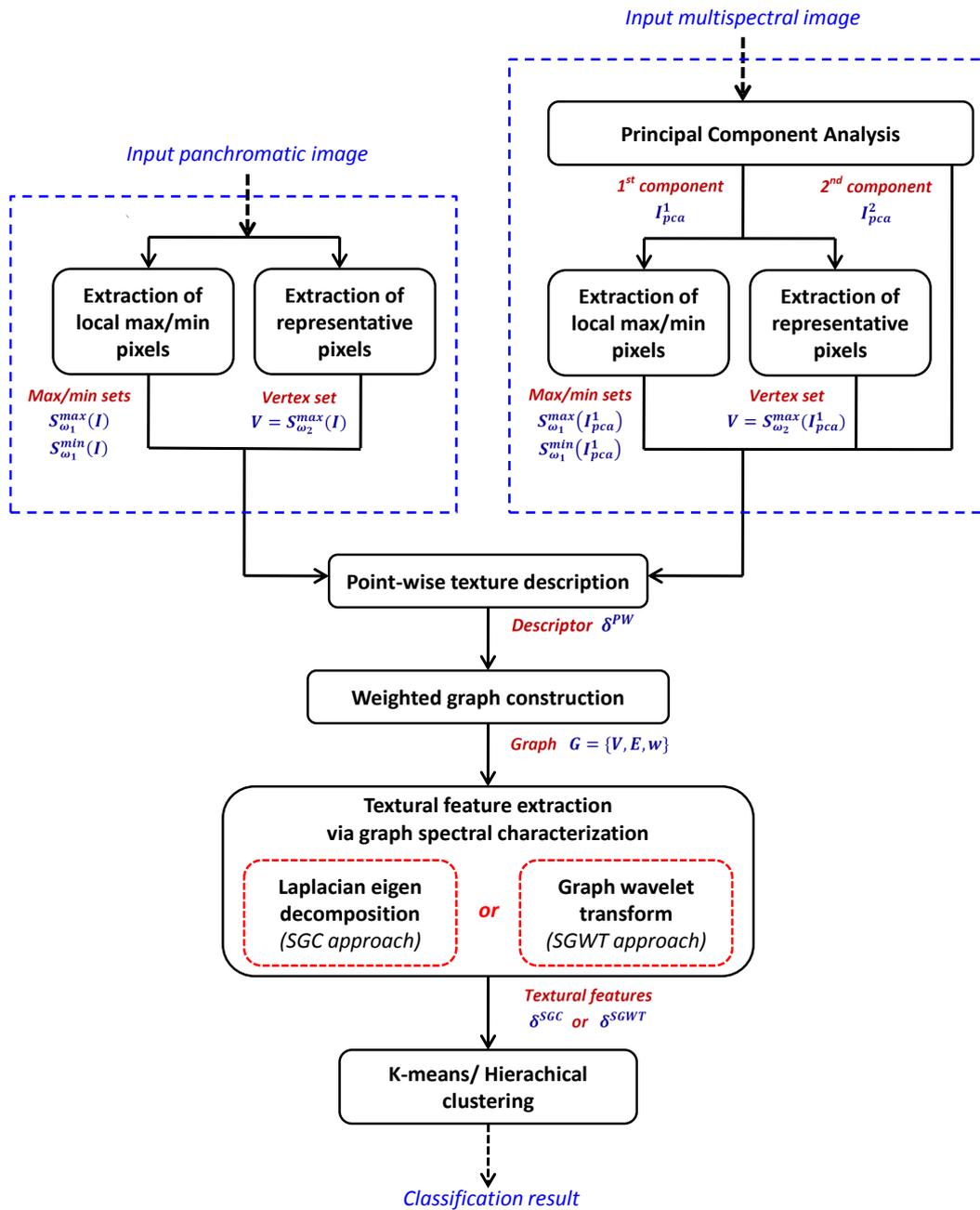


Figure 4.2 – Proposed framework for VHR panchromatic/multispectral image texture characterization applied to classification task using a pointwise graph-based approach.

be directly performed on the image. Next, a weighted graph \mathcal{G} is constructed to link the extracted keypoints based on their descriptors. Then, the spectral domain of graph \mathcal{G} is exploited to extract the final textural features. Here, two distinct approaches will be investigated and compared. They are the spectral graph clustering (SGC) approach based on the graph Laplacian eigen decomposition on the left-side red block and the spectral graph wavelet transform (SGWT) on the right-side one. For the final classification stage, unsupervised clustering algorithm is performed using these textural features.

For an input multispectral image, we remind that the spatial-spectral constraint needs to be taken into account. Here, we first perform the Principal Component Analysis (PCA) [61] on the image. The first and second PCA components are then used for extracting representative pixels and calculating their local pointwise descriptors. The other stages are similar to the case of panchromatic images. More details will be provided in the rest of this section.

4.3.1 Extraction of characteristic points for image representation and description

4.3.1.1 For panchromatic images

This stage involves the extraction of interest points for a sparse representation of textures as well as the extraction of local extrema (i.e. local maximum and local minimum pixels) that serve for computing local pointwise descriptors (Section 4.2). An important constraint is that representative pixels need to go along with the appearance of textures so that they can cover all textural zones from the image. Therefore in our scheme, we propose to exploit directly the local maxima (with the same or a different window size for their extraction) for this task. An advantage is that we do not need to search for another type of interest points. Moreover, these local maximum points always occur naturally within any radiometric variation, a basic property of texture, so that they are capable of covering all texture areas as required. From the figure, we distinguish the following sets:

- Two sets of extrema pixels $S_{\omega_1}^{\max}(I)$ and $S_{\omega_1}^{\min}(I)$ extracted using a search window $\omega_1 \times \omega_1$. These two sets are used to support the computation of pointwise descriptors.
- Another set of interest points $S = S_{\omega_2}^{\max}(I)$ for the keypoint-based texture representation and for graph construction stage (used as graph vertex set). Here, we can set $\omega_2 = \omega_1$ or $\omega_2 > \omega_1$ to speed-up the processing time of graph computation (i.e. the higher ω_2 is used, the coarser density of local max keypoints is obtained). It does not yield a coarser texture description result but a sub-sampled result.

Then, for each keypoint $p \in S$, the pointwise descriptor $\delta^{\text{PW}}(p)$ is extracted as in Equation (4.7). Hence, a set of descriptors is obtained for all keypoints in S and they will be exploited for the next stage of weighted graph construction.

4.3.1.2 For multispectral images

When the input image is not a panchromatic but a multispectral one, the Principle Component Analysis (PCA) is firstly applied, as observed in the right-side blue block of Figure 4.2. In the scope of multispectral image processing, the PCA technique has been exploited for many applications such as image compression, pansharpening, feature extraction, etc. Basically, it is used to convert a number of correlated bands from a multispectral image into a fewer number of uncorrelated bands, called PCA bands, which contain the majority of information from the original image. Our proposition is to exploit only the first and the second components. As observed in our experiments, the first PCA band, which captures most of the image content, acts as a panchromatic image while the second PCA image highlights the effect of the near-infrared band involving a crucial factor for vegetation recognition in remote sensing field. That is why we propose mainly using the first PCA image to extract representative pixels as well as local extrema to represent image textures. Then, the second PCA image, which provides complementary spectral information, is integrated into the computation of local pointwise descriptors

for multispectral images. In general, more than two first PCA images can be also considered to provide more complementary spectral information. However, the more PCA bands are used, the more computational time is required, not only for the calculation of local sparse descriptors but also for the later graph construction stage, hence the higher algorithm complexity will be resulted.

From the framework in Figure 4.2, after applying the PCA on the input multispectral image I and obtaining the PCA image denoted by I_{pca} . We propose to:

- 1) Use the first PCA image I_{pca}^1 as a panchromatic one to extract two sets of local extrema points $S_{\omega_1}^{\max}(I_{\text{pca}}^1)$ and $S_{\omega_1}^{\min}(I_{\text{pca}}^1)$, and the set of keypoints $S = S_{\omega_2}^{\max}(I_{\text{pca}}^1)$ (with $\omega_2 \geq \omega_1$) as in the previous case of panchromatic images.
- 2) Exploit both the first and the second PCA images, i.e. I_{pca}^1 and I_{pca}^2 , for the generation of pointwise descriptors.

In order to generate the pointwise feature vector $\delta^{\text{PW}}(p)$ for each keypoint $p \in S$, we firstly extract features from the first PCA image I_{pca}^1 as in Equation (4.7). Then, some features from the second PCA image I_{pca}^2 will be added to provide supplementary spectral information. Because such spatial features related to the distance and orientation (μ_d , σ_d^2 , \overline{R}_α and $D_{\alpha,\theta}$) are the same for both I_{pca}^1 and I_{pca}^2 , only intensity (i.e. spectral) features (μ_I , σ_I^2) calculated from I_{pca}^2 will be inserted into $\delta^{\text{PW}}(p)$.

As a result, the description vector constructed for multispectral case can be summarized:

$$\forall p \in S, \quad \delta^{\text{PW}}(p) = [\delta_1^{\text{PW}}(p), \delta_2^{\text{PW}}(p)], \quad (4.8)$$

where:

- $\delta^{\text{PW}}(p) \in \mathbb{R}^{16}$,
- $\delta_1^{\text{PW}}(p) \in \mathbb{R}^{12}$ is computed as (4.7) for I_{pca}^1 (as in case of a panchromatic image):

$$\delta_1^{\text{PW}}(p) = \left[\mu_{I_{\text{pca}}^1}^{\max}(p), \sigma_{I_{\text{pca}}^1}^{2\max}(p), \mu_d^{\max}(p), \sigma_d^{2\max}(p), \overline{R}_\alpha^{\max}(p), D_{\alpha,\theta}^{\max}(p), \right. \\ \left. \mu_{I_{\text{pca}}^1}^{\min}(p), \sigma_{I_{\text{pca}}^1}^{2\min}(p), \mu_d^{\min}(p), \sigma_d^{2\min}(p), \overline{R}_\alpha^{\min}(p), D_{\alpha,\theta}^{\min}(p) \right].$$

- $\delta_2^{\text{PW}}(p) \in \mathbb{R}^4$ includes only intensity features computed as (4.1), (4.2) for I_{pca}^2 :

$$\delta_2^{\text{PW}}(p) = [\mu_{I_{\text{pca}}^2}^{\max}(p), \sigma_{I_{\text{pca}}^2}^{2\max}(p), \mu_{I_{\text{pca}}^2}^{\min}(p), \sigma_{I_{\text{pca}}^2}^{2\min}(p)].$$

Since $\delta^{\text{PW}}(p)$ contains independent features from different categories (intensity, distance and angle), the range of their values should be standardized. We propose to normalize each type of features towards a centered reduced distribution. It means that the values of each one have zero-mean and unit-variance. This normalization step is required for the calculation of distance measure between description vectors serving for the next graph construction stage.

4.3.2 Weighted graph construction

A weighted graph \mathcal{G} is now constructed to connect all the extracted keypoints from the keypoint set S . For a reminder, $S = S_{\omega_2}^{\max}(I)$ in case of input panchromatic image and $S = S_{\omega_2}^{\max}(I_{\text{pca}}^1)$ in case of multispectral image. In this work, edge weights of \mathcal{G} are computed based on the vertex

similarity in terms of local feature descriptors. Given that each vertex has itself a description vector, graph edges can be identified by two general ways. The first one is to use a threshold to preserve only strong edges between vertices while the second is to assign each vertex to its k -closest neighbors in terms of distance between their description vectors (i.e. k -strongest edges). This issue has been discussed in Section 3.5 (Chapter 3). The second way is chosen in this work as it seems to be simpler to control the number of edges for each vertex than to fix a threshold for eliminating weak edges.

Thus, denote $\mathcal{G} = \{V, E, w\}$ the constructed graph, we have:

- Vertex set:

$$V = S = \begin{cases} S_{\omega_2}^{\max}(I) & \text{for panchromatic images,} \\ S_{\omega_2}^{\max}(I_{\text{pca}}^1) & \text{for multispectral images.} \end{cases} \quad (4.9)$$

- Edge set:

$$E = \{(p, q); p, q \in V, q \in \mathcal{N}_k(p) \vee p \in \mathcal{N}_k(q)\}. \quad (4.10)$$

- Weight function:

$$w(p, q) = e^{-\gamma[\text{dist}(\delta^{\text{PW}}(p), \delta^{\text{PW}}(q))]^2}, \forall (p, q) \in E, \quad (4.11)$$

where $\text{dist}(\delta^{\text{PW}}(p), \delta^{\text{PW}}(q))$ represents the distance measure between normalized pointwise description vectors of two vertices p and q . For the actual work, the Euclidean distance $\|\delta^{\text{PW}}(p) - \delta^{\text{PW}}(q)\|_2$ is employed. $\mathcal{N}_k(p)$ (resp. $\mathcal{N}_k(q)$) denotes the set of k -closest neighbors of p (resp. q) in terms of description vector distances and γ is a free parameter fixed to 1 in our implementation. Therefore, we have:

$$w(p, q) = e^{-\|\delta^{\text{PW}}(p) - \delta^{\text{PW}}(q)\|_2^2}, \forall (p, q) \in E. \quad (4.12)$$

4.3.3 Textural feature extraction

Once the texture graph \mathcal{G} has been constructed, the final textural features are derived from the spectral domain of graph \mathcal{G} . As previously mentioned, we employ two techniques: the spectral graph clustering (SGC) and the spectral graph wavelet transform (SGWT).

4.3.3.1 Via SGC approach

Spectral graph theory [44] proves that the eigenvectors of the graph Laplacian matrix have the capacity to carry information for vertex characterization and distinction. This aspect becomes the fundamental principle of the spectral graph clustering [45, 46] which has been reviewed in Section 3.2.3 (Chapter 3). Since the notion of textures has been embedded into our own graph \mathcal{G} , the Laplacian eigenvectors can reflect a texture representation space. According to this approach, the first C eigenvectors corresponding to the C smallest eigenvalues of the graph Laplacian matrix are directly exploited to form textural features. We note that C is the number of clusters (i.e. classes) expected from the clustering process.

It should be noted that a full eigen-decomposition of the Laplacian matrix with high computational complexity is not necessary. Because our texture graph is sparsely constructed and only the first C eigenvectors are required, approximation methods can be used such as the application of Polynomial Filters [62] which is implemented in MATLAB.

In short, let $\delta^{\text{SGC}}(p)$ be the texture characterization vector of vertex p using the spectral graph clustering approach:

$$\forall p \in V, \delta^{\text{SGC}}(p) = [\chi_0(p), \chi_1(p), \dots, \chi_{C-1}(p)], \quad (4.13)$$

where χ_i is the $(i + 1)^{\text{th}}$ eigenvector of the graph Laplacian matrix. For more details about the graph Laplacian matrix and the principle of SGC technique, readers are invited to review Section 3.2 (Chapter 3).

4.3.3.2 Via SGWT approach

This approach will not directly employ the Laplacian eigenvectors as the SGC technique. Indeed, the contribution of each eigenvector will be manipulated using low-pass and band-pass filters as mentioned in Section 3.3.2 (Chapter 3). Here, the spectral graph scaling function coefficients and spectral graph wavelet coefficients at different scales are considered as textural features.

Since the computation of SGWT coefficients requires the function on vertices $f \in \mathbb{R}^{|V|}$, we propose to use the vertex intensities. In case of multispectral images, both intensity values from the first and the second PCA images are considered. In short, let $\delta^{\text{SGWT}}(p)$ be the texture characterization vector of vertex p using the graph wavelet transform approach. For panchromatic images, we have:

$$\forall p \in V, \delta^{\text{SGWT}}(p) = [S_f(p), W_f(s_1, p), \dots, W_f(s_J, p)]. \quad (4.14)$$

where $S_f(p)$ and $W_f(s_j, p)$ respectively denote the scaling function coefficient and the wavelet coefficient at scale s_j computed for vertex p . They are generated using Equations (3.15) and (3.16). Then, in case of multispectral images, the vector becomes:

$$\forall p \in V, \delta^{\text{SGWT}}(p) = [\delta_1^{\text{SGWT}}(p), \delta_2^{\text{SGWT}}(p)], \quad (4.15)$$

in which:

- $\delta_1^{\text{SGWT}}(p) = [S_f(p), W_f(s_1, p), \dots, W_f(s_J, p)]$, computed for the image I_{pca}^1 ,
- $\delta_2^{\text{SGWT}}(p) = [S_f(p), W_f(s_1, p), \dots, W_f(s_J, p)]$, computed for the image I_{pca}^2 .

For the implementation of this approach, although the computation of $S_f(p)$ and $W_f(s_j, p)$ involves all Laplacian eigenvectors, we do not need to perform the full eigen-decomposition. In [50], the authors propose that SGWT can be approximately implemented by using the truncated Chebyshev polynomial approximation [63]. Indeed, from the constructed graph \mathcal{G} and a given number of scales J , following steps are activated for the implementation of SGWT (see [50] for more details):

1. Calculate the graph Laplacian matrix \mathcal{L} and estimate the upper bound λ_{max} of the graph spectral domain.
2. Calculate the set of scales $\{s_j\}_{j=1, \dots, J}$ which decides different levels of stretching filter kernels on the frequency domain $[0, \lambda_{\text{max}}]$. Then, design the set of filters comprising one low-pass $h(\cdot)$ and J band-pass $g(s_j \cdot)$ corresponding to each scale s_j .
3. Compute the truncated Chebyshev polynomial approximation as describing in [50].

4. For each vertex $p \in V$, compute all SGWT coefficients including the scaling function coefficient $S_f(p)$ and J wavelet coefficients $\{W_f(s_j, p)\}_{j=1, \dots, J}$ by supposing that the function on vertex $f(p)$ is the intensity $I(p)$ (for panchromatic images); or the intensity values of the first and the second PCA images, i.e. $I_{\text{pca}}^1(p)$ and $I_{\text{pca}}^2(p)$ (for multispectral images).

4.3.4 Clustering algorithm for unsupervised classification

The final stage of the proposed framework is to perform a clustering algorithm for unsupervised classification purpose. In fact, this stage is proposed to evaluate the performance of the extracted textural features (via SGC and SGWT approaches). Here, we note that the texture-based classification is sparsely carried out on the keypoint set (i.e. graph vertices). Certain clustering methods can be considered such as the *K-means algorithm* [47] or the *Hierarchical algorithm* [64]. Texture characterization vectors from (4.13), (4.14) or (4.15) are ready to be exploited as inputs of those algorithms, regarding to the SGC and the SGWT approach, respectively.

4.4 Experimental results

In this section, some experiments are carried out to evaluate and validate the effectiveness of the proposed texture-based classification framework. We first describe the input multispectral image data and the experimental setup in Sections 4.4.1 and 4.4.2. Then, the algorithm performance will be evaluated and compared to reference methods in terms of classification accuracy (Section 4.4.3) as well as computational time (Section 4.4.4). We finally study the sensitivity of the method to its parameters in Section 4.4.5.

4.4.1 Datasets

Data used for the experimental study in this chapter are the multispectral images acquired by the VHR Pléiades Constellation (PHR), copyright CNES©, the French Space Agency. We exploit the 4-band multispectral pan-sharpened products with 50cm spatial resolution, available from *www.astrium-geo.com*. These images were originally acquired using the satellite’s high resolution panchromatic sensor (70cm at nadir, resampled at 50cm) and its 4-band multispectral sensor with lower resolution (resampled at 2m) including red (R), green (G), blue (B) and near infrared (IR) components. Pan-sharpening process was then applied to combine these raw panchromatic and multispectral sources to generate the 50cm high resolution 4-band multispectral products. Two following datasets are used for our experiments.

Dataset 1 A semi-urban area is studied with the appearance of forest, flat vegetation and different man-made structures, etc. from a 4-band Pléiades image acquired in Naples, Italy (February 2013), 8 bit coded, 50cm resolution. The experimental ROI (region of interest) consists of 600×800 pixels, shown in Figure 4.3(a).

Dataset 2 The evolution of different trees in a boundary forest zone is investigated from another 4-band Pléiades image acquired in Colorado, USA (August 2013), 12 bit coded, 50cm resolution. The ROI patch size is 600×600 pixels, shown in Figure 4.4(a).

4.4.2 Experimental setup

We perform unsupervised classification experiments on our two datasets. The experimental procedure is triggered as the processing framework presented in Figure 4.2. In order to evalu-

ate the performance of our proposed method, some reference approaches are implemented for a comparative study. We propose to investigate the naive Intensity-based approach (IB), the Local Neighborhood (LN) as well as the Cooccurrence Matrix technique with Haralick features (GLCM). Also, as our proposition involves pointwise local feature description and graph modeling, we also propose to embed each of these reference methods into the graph model to yield graph-based textural features for an equivalent comparison.

4.4.2.1 Reference methods

For both data sets, the following implementations are carried out. We note that the first and second PCA images (I_{pca}^1 and I_{pca}^2) will be exploited for all computations.

Only reference methods

- *Intensity-based (IB)*: directly use vertex intensity values (from both I_{pca}^1 and I_{pca}^2).
- *Local Neighborhood (LN)*: employ all pixel intensities of a $\omega_{ln} \times \omega_{ln}$ dense neighborhood around each vertex. Since, the intensities from the two PCA images are exploited, the dimension of LN feature vector will be $2 \times \omega_{ln}^2$.
- *GLCM*: from the $\omega_{glcm} \times \omega_{glcm}$ neighborhood around each vertex, compute 4 cooccurrence matrices at 4 directions (0° , 45° , 90° and 135°), then extract 5 textural Haralick features from each matrix including the *contrast*, *correlation*, *homogeneity*, *energy* and *entropy* in order to create a 20-D feature descriptor from each PCA image. Hence, the length of the final vector for each keypoint is 40.

Reference methods + Graph model

- *IB + Graph*: use IB descriptors to construct the weighted graph \mathcal{G} instead of our pointwise descriptors (ref. Section 4.3.2). Textural features are then generated via SGC and SGWT approaches, similar to the proposed strategy.
- *LN + Graph*: use LN descriptors to construct the graph \mathcal{G} .
- *GLCM + Graph*: use GLCM descriptors to construct the graph \mathcal{G} .

Proposed strategy

- *Only pointwise (PW) descriptor*: directly exploit the pointwise texture descriptor δ^{PW} in Equation (4.8) without graph construction.
- *PW+SGC*: our first proposition, use the pointwise descriptor with the spectral graph clustering approach to obtain the descriptor δ^{SGC} as in Equation (4.13).
- *PW+SGWT*: our second proposition, use the pointwise descriptor with the spectral graph wavelet transform to obtain the descriptor δ^{SGWT} as in Equation (4.15).

In addition, quantitative results compared with ground truth may be necessary for a better assessment of the algorithm performance. Because of not having the real ground truths of the studied scenes, we manually prepare them for both data sets, bringing the best land-cover interpretation of each studied scene in our point of view. These manual classified ground truths are displayed in Figure 4.3(b) for Dataset 1 and in Figure 4.4(c) for Dataset 2.

4.4.2.2 Parameter setting

Table 4.1 – Parameter setting in our implementation. Classification results are shown in Figures 4.3 and 4.4.

Process	Dedicated parameters	Name	Dataset 1 600 × 800	Dataset 2 600 × 600
PW	Window size for max/min extraction	ω_1	5 × 5	5 × 5
	→ Number of extracted local maxima	$ S_{\omega_1}^{\max} $	16591	9225
	→ Number of extracted local minima	$ S_{\omega_1}^{\min} $	19870	9376
	Window size for vertex extraction	ω_2	7 × 7	5 × 5
	→ Number of graph vertices	$ V = S_{\omega_2}^{\max} $	8340	9225
	Number of closest extrema for each vertex	K	20	20
LN	Neighborhood window size	ω_{ln}	7 × 7	7 × 7
	→ LN vector length	$2 \times \omega_{ln}^2$	98	98
GLCM	GLCM window size	ω_{glcm}	41 × 41	41 × 41
	→ GLCM vector length	2×20	40	40
Clustering	Number of classes	N_c	6	5
Graph modeling	Number of edges preserved for each vertex	$k \approx 1\% V $	100	100
	Number of first eigenvectors (SGC approach)	$C = N_c$	6	5
	→ SGC vector length	C	6	5
	Number of wavelet scales (SGWT approach)	J	3	3
	→ SGWT vector length	$2 \times (J + 1)$	8	8

Table 4.1 provides a list of parameters and their values which are set during our implementation. The obtained classification results are shown later in Figure 4.3 and Figure 4.4. In fact, multiple experiments are carried out with parameter tuning. Here, the table only presents the values which produce the best performance from our tests. Firstly, a window of size 5×5 pixels ($\omega_1 = 5$) is employed for searching the local max and local min pixels from both datasets. Since Dataset 1 is bigger and contains more texture variations than Dataset 2, more extrema pixels are detected. For the extraction of representative pixels, we keep the same window size ω_2 equal to 5×5 for Dataset 2 but increase it to 7×7 for Dataset 1 (to obtain fewer points and accelerate the processing time). Therefore, the numbers of graph vertices for Dataset 1 and Dataset 2 are 8340 and 9225, respectively. To obtain an automatic graph construction process, about 20 closest extrema of each type ($K = 20$) are used for computing pointwise descriptors (Section 4.2) and about 1% of edges are preserved for each vertex.

4.4.3 Classification results

Figure 4.3 shows the classification results of different methods applied to Dataset 1 while Table 4.2 provides the total computational time (TCT), the overall classification accuracy (OCA) as well as the *Kappa* coefficient (κ) of each method. Our main purpose of using OCA and *Kappa* is to perform comparative results with other methods by quantitative assessment, as the visualization of classification results can not be considered to be convincing. Here, OCA is defined as the percentage of keypoints (i.e. graph vertices) which are correctly classified according to labeled keypoints from the manual ground truths. In addition, the *Kappa* coefficient provides a measure of agreement between model classification and reality, and it can be calculated from the confusion matrix as proposed in [65]. Similarly, experimental results for Dataset 2 are shown in Figure 4.4 and Table 4.3.

From Tables 4.2 and 4.3, the best classification performance for both datasets comes from the proposed pointwise descriptor combined with graph model. Depending on each studied scene, PW+SGC or PW+SGWT can achieve a better OCA (and also *Kappa*) than the other. Here, PW+SGC produces a better performance for Dataset 1 with an OCA of 72.39% ($\kappa = 0.6273$) compared with 62.23% ($\kappa = 0.5123$) yielded by PW+SGWT. On the contrary, PW+SGWT attains the best performance with 91.52% ($\kappa = 0.8682$) for Dataset 2 while PW+SGC stops at 84.31% ($\kappa = 0.7606$). In both cases, we observe the increase in OCA and *Kappa* for all approaches when they are embedded into graph model. For example, OCA of GLCM method is enhanced from 42.59% to 48.02% for Dataset 1 and also from 48.81% to 55.72% for Dataset 2. Similar behavior is found from IB and NL methods with an increase of 4.36% and 7.34% (for Dataset 1) as well as 10.7% and 8.93% (for Dataset 2), respectively. This issue confirms the crucial intervention of graph to encode the interaction and inter-connection among features within the image. Let us now investigate the performance in land-cover classification for each data set.

4.4.3.1 Result on data set 1

Table 4.2 – Comparison of total computational time (TCT), overall classification accuracy (OCA) and *Kappa* coefficient for Dataset 1.

Method	Description vector length	TCT	OCA	Kappa
IB	2	3.74s	45.54%	0.3391
LN	98	3.89s	53.28%	0.4250
GLCM	40	35.06s	42.59%	0.2740
PW	16	31.86s	55.31%	0.4227
IB+SGC	6	69.57s	49.90%	0.3716
LN+SGC	6	87.53s	60.62%	0.4921
GLCM+SGC	6	113.42s	48.02%	0.3542
PW+SGC	6	104.01s	72.39%	0.6273
PW+SGWT	8	99.81s	62.23%	0.5123

The first remark is that the obtained OCA and *Kappa* coefficient from all approaches are not very high due to the presence of shadow in the image which causes a considerable number of misclassified points. However, as observed in Figure 4.3(k) and Figure 4.3(l) compared to the manual ground truth in Figure 4.3(b), both PW+SGC and PW+SGWT provide very good results in terms of texture characterization and discrimination according to the selected keypoints from the image. A thematic coherence is found here for the observed scene. First of all, the structured building (marked in red in the ground truth) can be distinguished from other buildings (in yellow) only by the GLCM method (Figure 4.3(f) and Figure 4.3(i)) or by our proposed methods (Figure 4.3(k) and Figure 4.3(l)). However, the GLCM approach produces very poor results for other classes with a lot of mixtures between them, hence obtains quite low OCA. Secondly, resulted classes from the proposed method appear homogeneous and well separated, providing a general discrimination and good representation of different classes in the scene. Meanwhile, the results of IB, LN or GLCM are not capable to do so since there exist some intermediate classes (the red class in Figure 4.3(e), the yellow one in Figure 4.3(f), etc.) whose nature can not be defined.

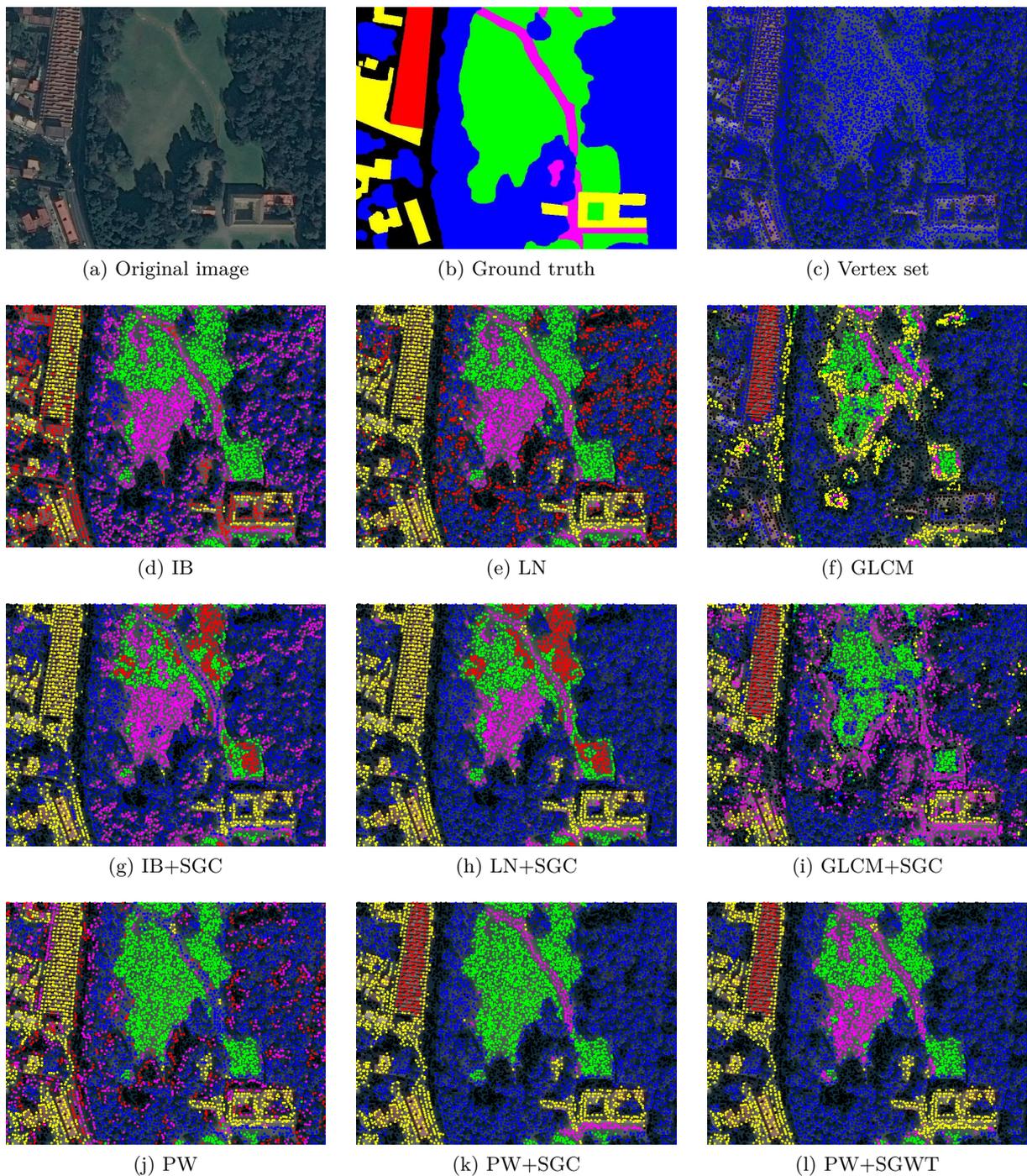


Figure 4.3 – Classification results for Dataset 1: (a) Original image patch (600×800 pixels); (b) Manual ground truth; (c) Vertex set (i.e. extracted local maxima); (d)-(e)-(f) Three reference methods; (g)-(h)-(i) Reference methods combined with SGC approach; (j) Pointwise descriptor (PW) without graph; (k) The proposed PW+SGC method; (l) The proposed PW+SGWT method.

Within the proposed method, the main difference between PW+SGC (Figure 4.3(k)) and PW+SGWT approach (Figure 4.3(l)) is found from their behavior on the transition between 2 classes: the green grass and the magenta soil/road. Indeed, the computation of PW+SGWT in

(4.15) involves the contribution of the first and second PCA image intensities (I_{pca}^1 and I_{pca}^2). Thus, more points are classified in the magenta class due to the affect of I_{pca}^2 . Compared to the proposed manual ground truth, a better result comes from PW+SGC approach thanks to its higher OCA (72.39% against 62.23%) as well as $Kappa$ (0.6273 against 0.5123). However, in terms of land-use classification, the result in Figure 4.3(k) may be interesting as well. Like other approaches, the drawback of the proposed method is caused by the presence of strong shadows where textures are not robust enough to deny them. More thematic features need to be considered to overcome this problem.

4.4.3.2 Result on data set 2

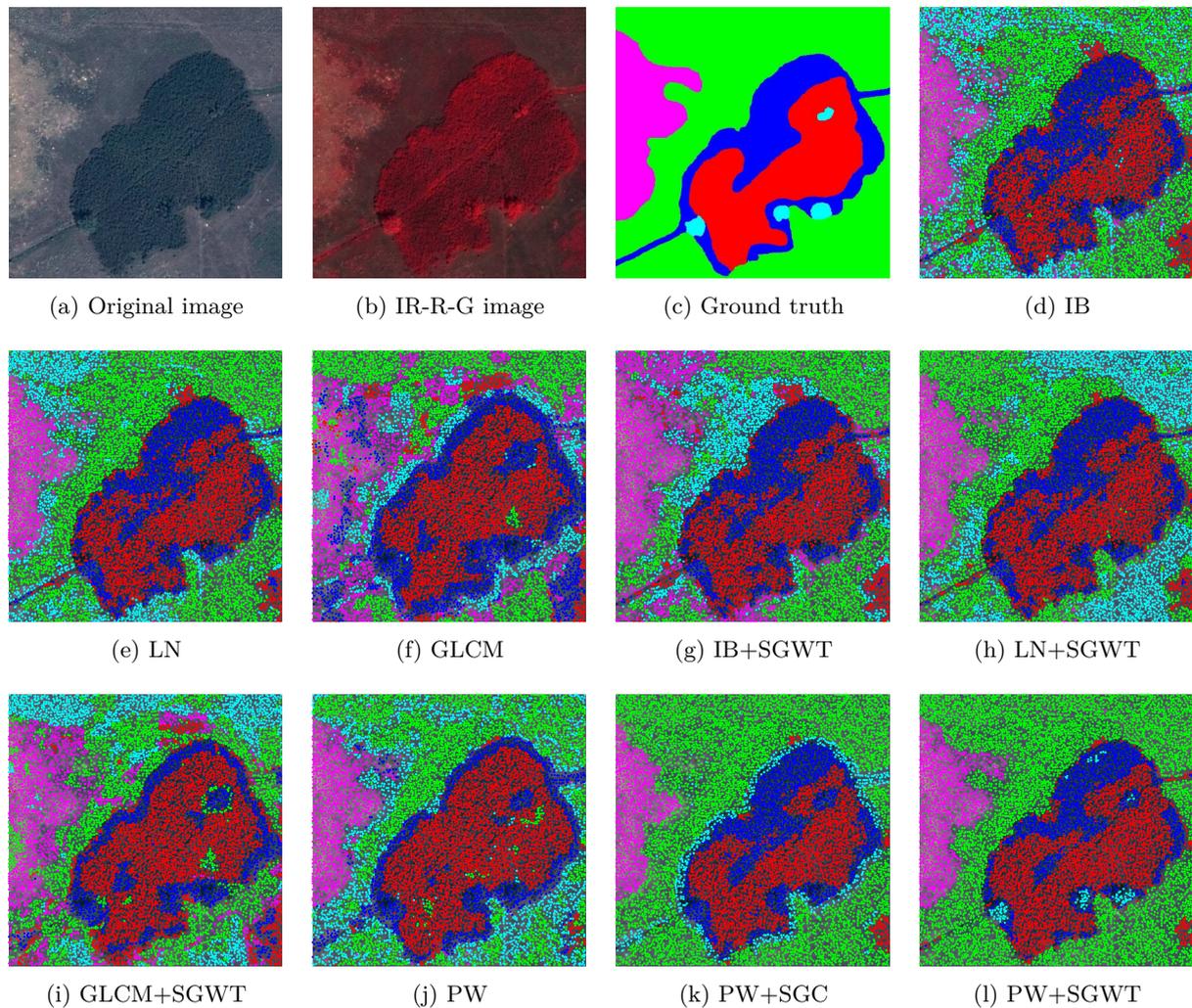


Figure 4.4 – Classification results for Dataset 2: (a) Original image patch (600×600 pixels); (b) IR-R-G display for better vegetation distinction; (c) Manual ground truth; (d)-(e)-(f) Three reference methods; (g)-(h)-(i) Reference methods combined with SGWT approach; (j) Pointwise descriptor (PW) without graph; (k) The proposed PW+SGC method; (l) The proposed PW+SGWT method.

Without the affect of strong shadows, most of the obtained OCA and $Kappa$ coefficient for this dataset are generally higher than those from the previous one. Consequently, we attain the best OCA of 91.52% ($\kappa = 0.8682$) yielded by the PW+SGWT approach. As mentioned

Table 4.3 – Comparison of total computational time (TCT), overall classification accuracy (OCA) and *Kappa* coefficient for Dataset 2.

Method	Description vector length	TCT	OCA	Kappa
IB	2	2.59s	61.34%	0.5044
LN	98	2.69s	66.60%	0.5527
GLCM	40	37.90s	48.81%	0.3468
PW	16	22.56s	63.92%	0.5153
IB+SGWT	8	81.87s	72.04%	0.6149
LN+SGWT	8	117.31s	75.53%	0.6565
GLCM+SGWT	8	122.62s	55.72%	0.4116
PW+SGC	5	109.52s	84.31%	0.7606
PW+SGWT	8	105.17s	91.52%	0.8682

before, we would like to study the evolution of different vegetation types inside the forest zone. The ground truth in Figure 4.4(c) is manually generated with the help of the IR-R-G image in Figure 4.4(b) (using infrared, red and green as three color bands). It consists of 2 different classes of ground and 3 classes of trees. Visually and quantitatively, the best classification result is obtained by the proposed PW+SGWT approach shown in the Figure 4.4(l) where all the 5 expected classes are identified. A remarkable success of this approach is its capacity of discriminating some specific trees (colored in cyan) which occur very locally within other classes. This issue confirms the ability of the proposed method to deal with local textural features. The PW+SGC approach (Figure 4.4(k)), which has shown the best reaction for Dataset 1, can not distinguish those cyan trees because it pays more attention on the transition between forest and non-forest zones. However in general, both 2 proposed approaches provide fairly good classification results coherent to the observed scene.

Other comparative approaches are not efficient for this scene. The one that should be mentioned is the GLCM technique which obtains an OCA of only 48.81% ($\kappa = 0.3468$) in Figure 4.4(f) and 55.72% ($\kappa = 0.4116$) in Figure 4.4(i) when combined with graph. Indeed, this approach mixes the classes and is not able to provide a general discrimination among them. This failure was previously anticipated due to the fact that GLCM always employs dense neighborhoods in VHR images without any guarantee of stationary or locally stationary conditions.

4.4.4 Performance in time consumption

A comparison of total computational time (TCT) of all methods applied to each dataset can be found in Tables 4.2 and 4.3. We note that all implementations are performed using MATLAB on a machine *Xeon 3.6GHz, 16GB RAM*. According to both tables, the first observation is that the proposed PW+SGC and PW+SGWT methods averagely consume about 102s (for Dataset 1) and 107s (for Dataset 2) but provide very good classification results. So, they can be considered to have good performance compared with reference methods.

Secondly, there is an important increase in TCT when each method is combined with graph model, due to the complexity of graph construction algorithm. We now observe Table 4.4 where the detailed computational time of each stage from the proposed method is shown. This costly graph construction stage consumes about 70% to 80% of TCT, for both datasets. As mentioned, this complexity mainly depends on the number of graph vertices $|V|$ and the number

Table 4.4 – Detailed time consumption of the proposed framework.

Processing stage	Dataset 1	Dataset 2
PCA	0.16s	0.11s
Extraction of local extrema	3.58s	2.48s
Vertex pointwise description	28.12s	19.97s
Graph construction	66.93s	81.75s
Feature extraction via SGC	5.22s	5.21s
Feature extraction via SGWT	1.02s	0.86s
Total time		
<i>SGC approach</i>	104.01s	109.52s
<i>SGWT approach</i>	99.81s	105.17s

of edges k preserved for each vertex. Since we fix $k \approx 1\%|V|$ for an automatic sparse graph construction, the complexity now significantly involves in $|V|$. This aspect emphasizes the efficiency of the proposed method in terms of time consumption for large-size image data because the computational time depends on the number of extracted characteristic points and not on the image size.

4.4.5 Sensitivity of the proposed method

4.4.5.1 Sensitivity to the window size ω_2

Experiments are carried out to investigate the sensitivity of the proposed algorithm to the search window size (ω_2) for extracting representative points (i.e. graph vertices) in terms of OCA and computational time for graph construction. Results produced by the PW+SGC approach performed on Dataset 1 are shown in Table 4.5. We observe that when ω_2 increases, graph construction time is considerably reduced. OCA is slightly enhanced which means that a good classification process is still ensured. However, the more we increase ω_2 , the fewer characteristic points are extracted for image representation and thus, more details may be lost from the image content. Therefore, this compromise needs to be taken into account when using such a pointwise approach.

Table 4.5 – Computational time for graph construction using different search window sizes for vertex extraction. Experiments are performed on Dataset 1.

Window size ω_2	Number of vertices $ V = S_{\omega_2}^{\max} $	Time	Number of correct classified points	OCA
5×5	16591	327.96s	11605	69.95%
7×7	8430	66.93s	6102	72.39%
9×9	4794	22.52s	3558	74.22%

4.4.5.2 Sensitivity to the number K

We now study the sensitivity of the method to another key parameter of the proposed strategy: the number of closest maxima and minima (K) used for vertex feature description. Experiments are conducted on Dataset 1 from which all other parameters are set as in Table 4.1 while K varies from 10 to 25. The first remark is that K is dedicated to the computation of vertex description vectors (Section 4.2). Hence, in terms of time consumption, varying K only differs

the computational time of the vertex description stage that we can observe in Table 4.6. Next, in terms of classification performance, it is certain that the PW approach itself is first affected. Also, as the proposed graph-based algorithm employs PW descriptors for graph construction, its performance will be influenced as well. These behaviors can be found in the last two columns of Table 4.6. We observe the best performance of the proposed PW+SGC method (72.39%) at $K = 20$. Experimental results show that increasing K will enhance the homogeneity and smoothness of classification results. However, if K is too high, some local textural zones become over-smoothed. For example, when K is equal or over 25, the soil/road (magenta class) in Figure 4.3(k) is occupied by the grass green class. That is the reason why the OCA of PW+SGC ($K = 25$) is decreased to 65.56% (Table 4.6).

Table 4.6 – Sensitivity of the proposed method to the number of closest extrema (K) used for vertex description. Experiments are performed using the PW and PW+SGC methods on Dataset 1.

Parameter K	Computation time		OCA	
	PW	PW+SGC	PW	PW+SGC
10	22.31s	98.20s	52.11%	63.99%
15	25.44s	101.33s	53.82%	70.41%
20	28.12s	104.01s	55.31%	72.39%
25	31.12s	107.01s	56.86%	65.59%

4.4.5.3 Sensitivity to clustering technique

According to the framework in Figure 4.2, both *K-means* and *Hierarchical* algorithms can be performed for the final clustering task. Our experimental results show that they compensate for each other but generally, the *K-means* technique produces better performance when the number of classes is well fixed. Meanwhile, the advantage of the *Hierarchical* algorithm is that when the number of classes changes, we observe only the phenomenon of merging or splitting from certain classes without affecting the others. Here, the classification results for our datasets shown in Figure 4.3 and Figure 4.4 are generated using the *K-means* algorithm since the number of classes has been fixed to be suitable to the manual ground truths with the best land-cover scene representation ($N_c = 6$ for Dataset 1 and $N_c = 5$ for Dataset 2, as in Table 4.1). However, in some other experiments, the *Hierarchical* clustering might be the preferred option.

4.5 Conclusion

In this chapter, a novel local texture characterization method applied to VHR optical image classification has been presented. In such a method, the pointwise (PW) descriptor is first developed by integrating spectral and spatial information of the local maxima and local minima extracted from the image. A similarity graph is then constructed to encode these descriptors and graph’s spectral characteristics are differently exploited to create a representation space of textural features. Experiments performed on VHR multispectral Pléiades images have demonstrated the effectiveness of the proposed pointwise descriptor as well as the crucial intervention of graph model. Unsupervised classification results provided by our framework are very promising and competitive in terms of good texture discrimination and land-cover interpretation compared to classical dense approaches.

Some perspective work based on this chapter will be discussed in Chapter 8. In the next chapter, we continue to develop another pointwise graph-based model to tackle texture tracking

task applied to the detection of land-cover changes as well as the detection of glacier displacements using SAR images.

Pointwise graph-based texture tracking for change detection from SAR images

Science never solves a problem without creating ten more.

George Bernard Shaw

5.1	Introduction	70
5.1.1	Context: change detection using SAR images	70
5.1.2	Motivation of the chapter	71
5.2	Texture tracking for change detection based on a pointwise graph-based approach	71
5.2.1	Proposed framework	71
5.2.2	Extraction of local extrema points adapted to SAR images	72
5.2.3	Weighted graph construction from keypoints	74
5.2.4	Change measure generation	75
5.3	Experiments	76
5.3.1	Data description and evaluation criteria	76
5.3.2	Change detection results	77
5.3.3	Sensitivity to parameters	81
5.3.4	Comparison with the NLM model	86
5.3.5	Analysis of the algorithm complexity	87
5.3.6	Exploitation of the local minimum pixels	89
5.4	Application to glacier displacement detection	90
5.4.1	Proposed method	90
5.4.2	Preliminary results	91
5.5	Conclusion	92

The content of this chapter is mainly based on the following published papers:

- M. T. Pham, G. Mercier, J. Michel, *Change detection between SAR images using a pointwise approach and graph theory*, IEEE Trans. Geosci. Remote Sens., vol. 54, issue 4, pp. 2020-2032, 2016.
- M. T. Pham, G. Mercier, J. Michel, *A keypoint approach for change detection between SAR images based on graph theory*, Proc. 8th Int. Wksh. Anal. Multitemp. Remote Sens. Images (Multitemp), Annecy, France, 2015, pp. 1-4.

5.1 Introduction

5.1.1 Context: change detection using SAR images

Synthetic Aperture Radar (SAR) has been extensively used in remote sensing field thanks to its capacity of acquiring data under any atmospheric and solar illumination conditions. One of the most significant applications is to exploit multitemporal SAR images for change detection task which serves for understanding and evaluating land-cover changes occurring after a natural or anthropic disaster, or for identifying and monitoring land-use development over time within certain agricultural, forestry and urban zones [66–69]. This chapter is dedicated to the context of change detection in multitemporal SAR images. We now review some state-of-the-art approaches in the related literature.

Many methods have been proposed to tackle the problem of image change detection. A systematic survey can be found in [70]. However, change detection using SAR images is still considered as a challenging task due to the image perturbation caused by speckle noise. This intrinsic multiplicative noise often increases the miss-detection and false alarm rates, hence considerably reduces the detection performance. Consequently, unlike in optical remote sensing, any change detection technique proposed using SAR images should take into account the existence of speckle [66].

In the scope of SAR image change detection, one of the most fundamental approaches is the mean-ratio detector (MRD) which is based on the ratio of local intensity means of pixel patches [71]. To reduce the effect of multiplicative speckle noise, many authors propose to employ the log-ratio detector (LRD) by performing logarithm operator on the ratio of local means [72, 73]. Next, some recent work has proposed to cooperate these two ratio operators to exploit their complementary information. In [74, 75], the authors perform the wavelet-based fusion technique on both the mean-ratio and log-ratio images in order to generate the final difference image (DI). In [76], the Gauss-log ratio DI is proposed by applying a Gaussian low-pass filter on pixel patches before performing the log-ratio operator. This DI is then fused with the log-ratio DI using wavelet transform to produce the final fused DI. In fact, these ratio operators are robust to speckle but they are limited to the comparison of first-order statistics since only the mean intensity of pixel patches is considered. Therefore, in case that the mean values of pixel patches stay the same, they are not able to detect changes, which probably increases the miss-detection rate.

To circumvent this drawback, other methods propose to consider higher order statistics of pixel patches by using the similarity measure of their local probability density functions (PDFs). In [30, 77], the authors study several approaches for PDF estimation and exploit the Kullback-Leibler divergence (KLD) as a distance measure for PDF comparison. As a result, they propose some local statistical measures for DI production including the Pearson-based KLD (PKLD) and the cumulant-based KLD (CKLD). These detectors can be generated based on the first four order statistics of pixel local patches [30]. The Pearson-based PDF estimation is characterized by two parameters and can hold eight different types of distribution (e.g. Gaussian, Gamma, etc.). Being more robust, the CKLD is derived from the Edgeworth series expansion and can deal with non-parametric PDFs. However, the CKLD is still dependent on the accuracy of Edgeworth approximation from which the estimated PDFs are not too far from Gaussian models. In [78], by using a modified KLD called Jeffrey divergence, the authors propose the projection-based Jeffrey detector (PJD) which preliminarily performs better than the CKLD in certain conditions. Nevertheless, all of these local statistics-based approaches are still constrained to

the evaluation of dense neighborhoods around image pixels. Hence, in order to achieve good detection performance, large-size windows are generally considered.

5.1.2 Motivation of the chapter

Taking into account the aforementioned issues, our motivation in this chapter is to avoid using large-size dense neighborhood around each pixel to measure its change level, which is usually considered by the above classical methods. We propose to exploit the interaction among characteristic points to develop a novel texture tracking method to tackle change detection task by embedding the pointwise approach into a weighted graph model. As discussed in Chapter 2, the local extrema keypoints are capable of capturing the image’s important radiometric and contextual information. Their interaction can be encoded by a graph model based on their similarity measures for which only small pixel patches around keypoints are required. More importantly, the constructed graph is able to characterize both intensity and geometry information from the image content (cf. Chapter 3). To this end, if a graph is constructed from one of the two images, the change level between them can be then measured from how much the information of the other image still conforms to that graph’s structure. This remark leads to our proposed texture tracking strategy for change detection.

The remainder of this chapter is organized as follows. We describe the proposed pointwise graph-based texture tracking approach applied to change detection in Section 5.2. Next, Section 5.3 provides our experimental study effectuated on real SAR image data. Change detection results using the proposed algorithm are compared to those yielded by some reference methods including the MRD, LRD, the KLD-based approaches and the nonlocal mean (NLM) model proposed in [79]. A detailed analysis of the algorithm complexity will be also provided. Section 5.4 presents our effort to apply the proposed texture tracking strategy to the detection of glacier displacements with some preliminary experimental results. Finally, Section 5.5 concludes the chapter.

5.2 Texture tracking for change detection based on a pointwise graph-based approach

5.2.1 Proposed framework

An outline of the proposed algorithm is highlighted in Figure 5.1. The first stage involves the extraction of a set of characteristic points, denoted by S_ω , from the first image I_1 to capture important information that represents the image content. Next, a weighted graph \mathcal{G} is constructed from these keypoints. In the final stage, change measures between the two images I_1 and I_2 at the extracted keypoints are generated with the help of the intrinsic structure of graph \mathcal{G} .

It is worth noting that the choice of one image between the two (I_1 or I_2) for graph construction is random. That means one can also generate a graph \mathcal{G}' to encode the image I_2 . Then, using a similar principle, change measure can be derived from how much I_1 conforms and adapts to the structure of \mathcal{G}' . Our experimentation shows that in both cases, we obtain similar detection performance of the proposed algorithm. Hence, without any loss of generality, the proposed strategy described in the rest of this chapter will exploit I_1 for graph construction stage, as in Figure 5.1.

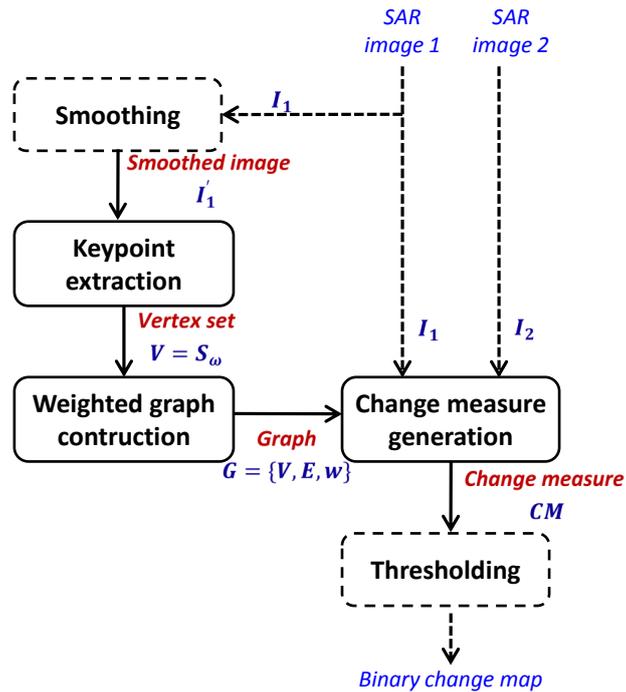


Figure 5.1 – Proposed framework for pointwise graph-based SAR image change detection.

5.2.2 Extraction of local extrema points adapted to SAR images

In Chapter 2, the local extrema (i.e. local minimum and local maximum pixels) have been proved to be able to capture and characterize the contextual information of the image from which they are extracted. They are now employed to perform such a pointwise texture tracking algorithm applied to change detection. Within this context, we propose to exploit the local maximum pixels and note that using the local minimum pixels also provides similar performance. Depending on the nature of changes, the choice of local max or local min keypoints may carry out slightly better performance than the other. In general, image regions with higher intensity involve more information than the lower ones. In these cases, the local maxima are more relevant. However, the local minima may be more suitable in case that one needs to focus on low-intensity regions such as roads, airports, rivers, etc. Our implementation mainly employs the local max keypoints to yield experimental results in this chapter. Then, in Section 5.3.6, some results are provided to confirm the similar performance coming from the use of local min keypoints.

In case that the image I contains a low level of noise (e.g. an optical image), the local maximum keypoints can be extracted directly using Equation (2.1) (Chapter 2) like we have done in Chapter 4. Here, the present chapter has to deal with SAR images which are perturbed by speckle noise. In order to account for its presence, the local max keypoints will be extracted from a smoothed version of I_1 , instead of the image itself. We note that this smoothing step is not a major task within the framework. Hence, for a fast implementation, a simple low-pass filter such as *Average*, *Median* or *Gaussian* can be used. Besides, one may wonder whether a despeckling filter is necessary for this stage, the response is that a low-pass filter should be more suitable, in our point of view. In fact, we expect a stationary filter which homogeneously smoothes the image in order to ensure a good distribution of keypoints for all image regions. Meanwhile, a despeckling filter tends to considerably suppress speckle in homogeneous zones but

effectively preserve fine details such as contours, corners, salient features, etc., hence induces an filtered image with a non-constant number of looks. Therefore, it may cause the fact that keypoints are extracted mostly on fine structures of the image but very sparsely on homogeneous zones, which is not expected in this study. To this end, our implementation employs a Gaussian filter for smoothing the image I_1 . In the rest of this chapter, we denote S_ω the keypoint set consisting of the local maximum pixels extracted from I'_1 , the smoothed version of I_1 , using the $\omega \times \omega$ search window. In summary, $S_\omega = S_\omega^{\max}(I'_1)$.

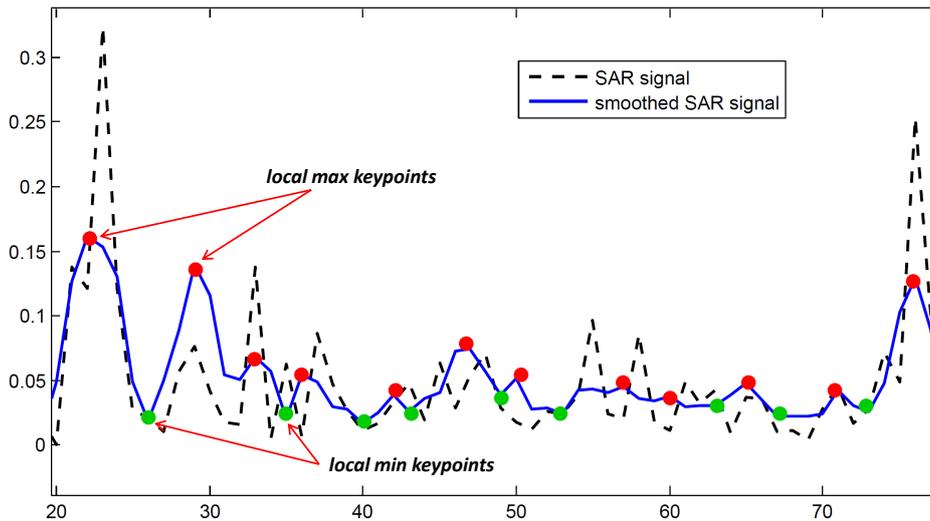


Figure 5.2 – Appearance of local max and min keypoints on SAR signal.

In Figure 5.2, we show an example of how the local maximum and local minimum keypoints appear within a SAR signal. Since the original signal (black dashed line) contains a lot of speckle noise, a smoothing procedure is first applied and the local extrema are detected from the smoothed signal (blue solid line). In case of two dimensional signals such as images, using the smoothed signal implicitly considers a pixel neighborhood (with the size of the smoothing filter), not each pixel itself, to detect local max and local min keypoints. Hence, the influence of speckle on each individual pixel is reduced.

Another question may arise concerning the possibility of using other types of feature points (Harris corner points, SIFT, SURF, etc. [27]) to perform our strategy. We remind that the main objective here is to detect land-cover changes. Hence, it is important to have keypoints covering all regions from the image so that we can ensure that all image zones will be considered for measuring changes. This issue has been discussed in Section 2.4 (Chapter 2), such feature points like Harris corners, SIFT or SURF may not be relevant since their appearance and distribution on the image plane usually focus on corners, edges and salient features. Hence, we might probably miss points from some homogeneous regions where changes can occur (increase miss-detection rate). One example can be found in Figure 5.3 where we compare the appearance of 2790 local extrema keypoints (Figure 5.3(a)) and 2790 Harris corner points (Figure 5.3(b)) detected from a SAR image. We observe that in Figure 5.3(b), there are some homogeneous regions that contain no points. Therefore, if these Harris corner points are used for change detection, we certainly miss changes from these regions. On the other hand, the local extrema points prove their capacity to cover all image zones involving any variation of intensity. They are more relevant for this application of change detection.

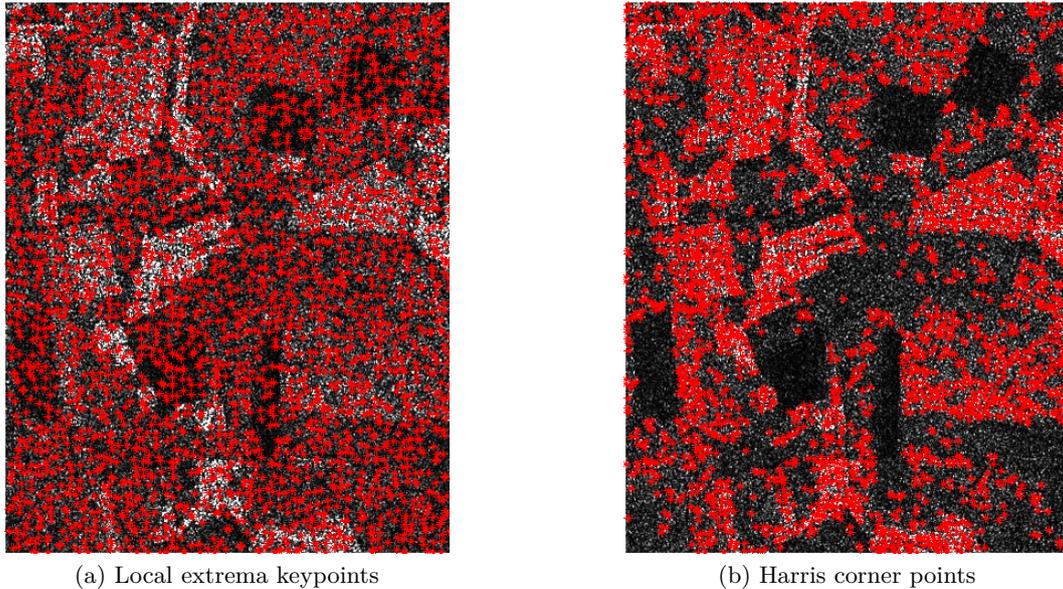


Figure 5.3 – Appearance of keypoints (red) extracted from a SAR image: (a) using the local extrema extractor; (b) using the Harris corner extractor.

Finally, it should be noted that our method does not detect changes based on the evolution of keypoints between two images, like the principle of change detection based on SIFT keypoints proposed by F. Dellinger et al. [80, 81]. In that work, the authors detect SIFT keypoints from both images and measure changes based on SIFT descriptors. Here, our local extrema keypoints are extracted only from one of the two images to construct a weighted graph. Then, change levels are derived from the irrelevance of the other image to the constructed graph structure.

5.2.3 Weighted graph construction from keypoints

The motivation to construct a weighted graph to connect keypoints has been described in Chapter 3, Section 3.5. In this work, the role of graph is to represent and encode the geometric structure of the first image regarding to its keypoint set. Denote $\mathcal{G} = \{V, E, w\}$ the weighted graph to be constructed, we define:

- Vertex set:

$$V = S_\omega, |V| = |S_\omega| = N, \quad (5.1)$$

- Edge set:

$$E = \{(p_n, p_k); p_n, p_k \in V, p_k \in \mathcal{N}_K(p_n) \vee p_n \in \mathcal{N}_K(p_k)\}, \quad (5.2)$$

where $\mathcal{N}_K(p_n)$ (resp. $\mathcal{N}_K(p_k)$) denotes the set of K -closest keypoints of p_n (resp. p_k) in terms of spatial distance on the image plane of I_1 .

- Weight function:

$$w(p_n, p_k) = e^{-\gamma[\text{dist}(p_n, p_k)]}, \forall (p_n, p_k) \in E, \quad (5.3)$$

where the term $\text{dist}(p_n, p_k)$ represents a distance measure of intensity between two vertices p_n, p_k and γ is a free parameter fixed to 1 in our implementation.

One can imagine that on the image plane of I_1 , each keypoint $p_n \in S_\omega$ now becomes a vertex and is connected to its K closest neighbors (i.e. in terms of spatial distance) by the graph \mathcal{G} . The edge weight $w(p_n, p_k)$ represents the similarity between two vertices p_n and p_k in terms of intensity. In general, this weight is generated via an exponential function of an intensity-based distance measure between p_n and p_k . However, it should be noted that in the scope of SAR image processing, a distance measure of intensity between two pixels is generally generated by taking into account the two pixel patches around them rather than themselves (due to the effect of speckle noise). For this task, several measures can be investigated such as distances based on the histogram of local patches, the ratio operator as well as the Kullback-Leibler divergence, etc. Here, we propose to employ the log-ratio (LR) distance of pixel patches around graph vertices. This operator has been proved to be robust to speckle in SAR imaging. Moreover, our experiments show that this measure is quite appropriate for small-size pixel patches which are expected in this work. Denote w_p the size of pixel patch considered for each vertex, the log-ratio distance can be defined:

$$\text{dist}(p_n, p_k) = \text{dist}_{\text{LR}}(p_n, p_k) = |\log \mu_1(p_n) - \log \mu_1(p_k)| = \left| \log \frac{\mu_1(p_n)}{\mu_1(p_k)} \right|, \quad (5.4)$$

where $\mu_1(p_n)$ (resp. $\mu_1(p_k)$) represents the mean intensity of the $w_p \times w_p$ pixel patch around vertex p_n (resp. p_k) from the first image I_1 .

As a result, edge weight between p_n, p_k is generated as follows:

$$w(p_n, p_k) = e^{-\gamma \left| \log \frac{\mu_1(p_n)}{\mu_1(p_k)} \right|}. \quad (5.5)$$

5.2.4 Change measure generation

Once the weighted graph \mathcal{G} has been constructed to connect all keypoints from S_ω , signal processing on the graph vertex domain mentioned in Section 3.3.1 (Chapter 3) can be applied. For this stage, we propose to exploit the graph adjacency matrix \mathcal{W} to perform the concentration of information at each vertex. We note that as discussed in the end of Section 3.3.1, the random walk matrix \mathcal{P} can also be used to take over this task. Since the graph \mathcal{G} is capable of characterizing the local geometric structure of the image I_1 at keypoint positions, our proposition is that the change level between I_1 and I_2 can be considered as the coherence and compatibility of radiometric information captured by the vertices of \mathcal{G} from these two images. In other words, the local texture structure at each keypoint from I_1 encoded by \mathcal{G} will be tracked on I_2 to search for the most coherent one. To perform this strategy, we define two following signals in \mathbb{R}^N :

$$\begin{aligned} f_1 &= [\log \mu_1(p_1), \log \mu_1(p_2), \dots, \log \mu_1(p_N)]^T, \\ f_2 &= [\log \mu_2(p_1), \log \mu_2(p_2), \dots, \log \mu_2(p_N)]^T, \end{aligned} \quad (5.6)$$

where $\mu_1(p_n)$ and $\mu_2(p_n)$ ($n = 1, \dots, N$) are defined as in Equation (5.4). They represent the mean intensity of the $w_p \times w_p$ pixel patch around the vertex p_n computed from the image I_1 and I_2 . Logarithm operator is proposed to theoretically reduce the influence of speckle.

The two signals f_1 and f_2 thus encapsulate the radiometric information of I_1 and I_2 , respectively, captured by the positions of graph vertices. According to Equation (3.7) (Chapter 3), the quantity $(\mathcal{W}f_1)(p_n)$ characterizes the radiometric information and geometric structure of I_1 with regard to the local topology of graph G at vertex p_n . Similar remark can be deduced for the quantity $(\mathcal{W}f_2)(p_n)$. Therefore, let $\text{CM}(p_n)$ be the change measure between two images at

keypoint p_n and $\|\cdot\|_1$ be the \mathcal{L}^1 norm, the proposed method defines:

$$\begin{aligned} \text{CM}(p_n) &= \|(\mathcal{W}f_1)(p_n) - (\mathcal{W}f_2)(p_n)\|_1 \\ &= \left| \sum_{p_k \sim p_n} w(p_n, p_k) \log \mu_1(p_k) - \sum_{p_k \sim p_n} w(p_n, p_k) \log \mu_2(p_k) \right| \\ &= \sum_{p_k \sim p_n} w(p_n, p_k) \left| \log \frac{\mu_1(p_k)}{\mu_2(p_k)} \right|. \end{aligned} \quad (5.7)$$

We observe that the generated change measure at each keypoint can be considered as a weighted summing of log-ratio distances derived from other keypoints thanks to the graph structure. This issue satisfies our expectation of taking advantage of weighted graph to encode the keypoint inter-connection. It also confirms the effectiveness of graph model when carrying out a pointwise approach. As previously mentioned, the random walk matrix \mathcal{P} could be also investigated to generate change measure. According to Equation (3.8) (Chapter 3), the calculation of $\text{CM}(p_n)$ by exploiting \mathcal{P} becomes:

$$\begin{aligned} \text{CM}(p_n) &= \|(\mathcal{P}f_1)(p_n) - (\mathcal{P}f_2)(p_n)\|_1 \\ &= \frac{1}{\sum_{p_k \sim p_n} w(p_n, p_k)} \sum_{p_k \sim p_n} w(p_n, p_k) \left| \log \frac{\mu_1(p_k)}{\mu_2(p_k)} \right|. \end{aligned} \quad (5.8)$$

In this case, the weighted summing of log-ratio measures from neighboring keypoints is normalized by dividing by the sum of weights. Hence, one may recognize certain similarity of the proposed method and the nonlocal mean (NLM) model presented in [79]. Indeed, our algorithm is basically distinct from the NLM method in the way of approaching and tackling change detection problem. More discussions will be provided to distinguish the two methods in Section 5.3.4.

5.3 Experiments

In this section, some experiments on real SAR image data are carried out in order to evaluate the performance and validate the effectiveness of the proposed algorithm. Reference detectors consisting of the MRD, LRD, the Gaussian-based KLD (GKLD), the cumulant-based KLD (CKLD) and the non-local mean (NLM) model are also implemented for comparative purposes. We first introduce two SAR data sets used in our experiments and some evaluation criteria. The experiment procedure is then described in details with several qualitative and quantitative results to evaluate the detection performance. As mentioned previously, a detailed comparison of the proposed method to the NLM model is addressed in Section 5.3.4. Then, an analysis of the algorithm complexity is provided. At the end, we investigate the algorithm performance in case of exploiting the local minimum keypoints for graph construction.

5.3.1 Data description and evaluation criteria

5.3.1.1 Data description

Data set 1 The first data set is a couple of 800×400 pixel images acquired by the RADARSAT-1 satellite in F2 and F5 mode before and after the eruption of the Nyiragongo volcano (January 2002) in Democratic Republic of the Congo. These images have a final spatial resolution of 10m and cover a semi-urban area of 8×4 km including the Goma international airport, Goma

city. Our experiments use the 8-bit coded version for the image intensity. The thematic ground truth change map between them is available for the quantitative evaluation of change detection performance. For visualization, the two input images and the ground truth mask are shown in Figures 5.4(a)-(b)-(c).

Data set 2 The second data set is a series of 10 multitemporal SAR images acquired within the same orbit by the RADARSAT-2 satellite from January to August, 2010. These 12-bit coded images have a spatial resolution of about 5m and cover an intensively agricultural area near the region of Chartres, France. They are mainly exploited for extracting vegetation indexes and studying different stages of crop growth for monitoring the agricultural surfaces within the area. For our change detection experiments, we employ the two images acquired in April 25 and in May 19 since they involve a strong land-cover change coming from vegetation developments. Since the entire image size is 10310×8234 pixels, a region of interest (ROI) of 1000×850 pixels is extracted from each image for the experiment. These studied ROIs are shown in Figures 5.6(a)-(b). No ground truth is available yet for this data set.

5.3.1.2 Evaluation criteria

For data set whose ground truth is available (data set 1), quantitative evaluation can be performed. We propose to plot the receiver operating characteristics curve (ROC) and to calculate the percentage of total errors (P_{TE}) as well as the percentage of overall accuracy (P_{OA}) for each detector. The ROC plots the good detection rate (P_{GD}) in function of the false alarm rate (P_{FA}) when the change measure is thresholded from the minimum to the maximum value [30]. For more details about the computation of P_{FA} , P_{GD} , P_{TE} and P_{OA} , readers are invited to consult Appendix A.

5.3.2 Change detection results

We perform change detection experiments on our two data sets. The experimental procedure is activated following the processing framework in Figure 5.1. For the extraction of characteristic points, our experiments employ a 3×3 pixel search window ($\omega = 3$) for both data sets to ensure a good density of keypoints for image representation. As a result, the numbers of extracted keypoints (i.e. graph vertices) are respectively 9545 for the first data set and 31356 for the second one. As mentioned, other change detection methods including the MRD, LRD, GKLD and CKLD are also implemented for a comparison against the proposed algorithm. A neighborhood window of 35×35 pixels is set for these reference methods. For our method, the graph \mathcal{G} is constructed with a number of $K = 50$ edges for each vertex and the patch size around each keypoint is set to 15×15 pixels ($w_p = 15$). It should be remarked that in our approach, the number K implicitly represents the size of a spatial neighborhood around each understudied keypoint in which we search for its closest neighbors within graph construction stage. In other words, K shows how far, on the image plane, the centered keypoint is connected to the others. We note that when K is fixed, the search neighborhood size dedicated to each keypoint is varied. Indeed, if the density of keypoints is high, the implicit search neighborhood size will be smaller, and vice versa. Experiments show that for $K = 50$, the average neighborhood size is estimated to be about 35×35 pixels. That is the reason why we choose a window size of 35×35 pixels for the above reference methods in order to ensure an equivalent comparison.

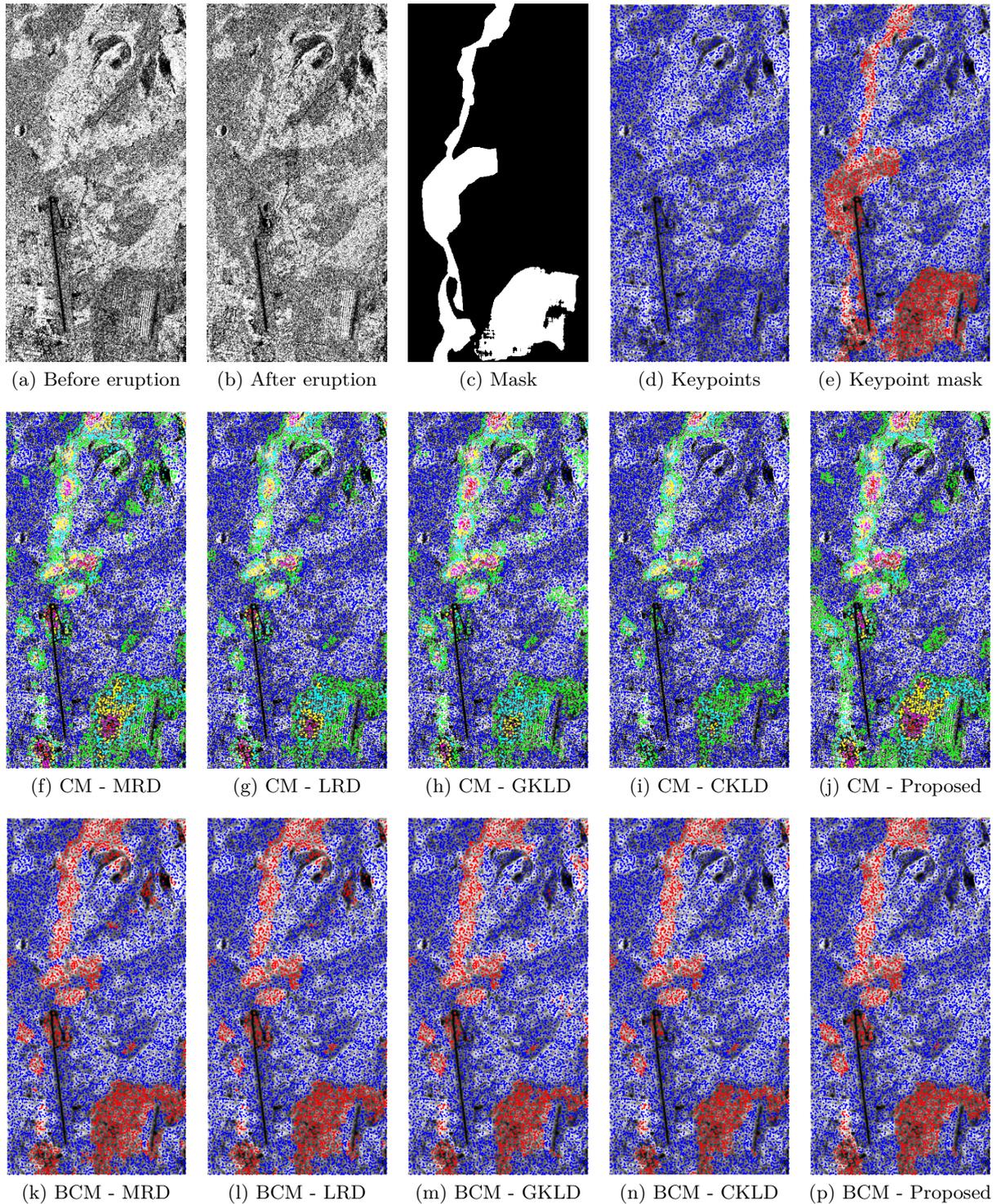


Figure 5.4 – Experimental study for Data set 1. (a) Image before volcano eruption. (b) Image after volcano eruption. (c) Ground truth change mask. (d) Extracted 9545 keypoints. (e) Ground truth change mask at keypoints including $N_c = 2011$ changed points (in red) and $N_u = 7534$ unchanged points (in blue). From (f) to (j) Change measures (CM) yielded by different detectors using color visualization. Change levels are set as follows: Blue < Green < Cyan < Yellow < Magenta < Red. From (k) to (p) Binary change maps (BCM) yielded by different detectors and thresholded using the *K-means* technique.

5.3.2.1 Results for data set 1

From Figure 5.4(f) to 5.4(j), pointwise change detection results for the first data set yielded by the proposed approach and by the four reference methods are performed. Change levels are encoded using color visualization from blue to red. Regarding to the ground truth mask at keypoints in Figure 5.4(e), the proposed method seems to provide the highest number of good detections but it also yields several false alarms. On the contrary, the CKLD gives the least number of false alarms but it ignores a lot of changed points, which certainly increases the miss-detection rate. The three remaining methods including MRD, LRD and GKLD produce quite similar behaviors which also involve the trade-off between the number of false alarms and missed detections.

For a better evaluation of change measure performance, Figure 5.5 shows the ROC plots for all detectors. We remind that the ROC is used to evaluate the quality of the change measure yielded by a detector by taking into account its false alarm rate and good detection rate regardless of the thresholding method. As observed in the figure, the proposed method (blue curve) considerably outperforms the other ones. This means that for any threshold value, our detection strategy is more efficient than reference methods in terms of good detection rate with respect to false alarm rate. Hence, the effectiveness of the proposed method is confirmed.

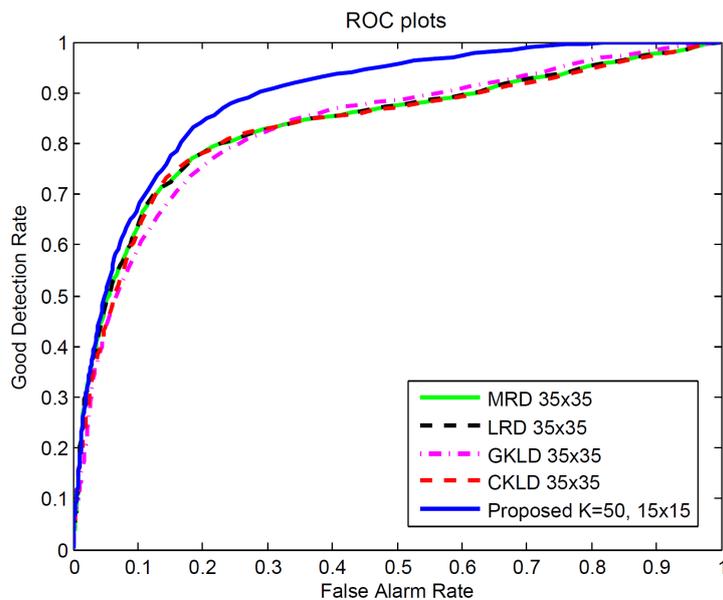


Figure 5.5 – ROC plots for the proposed change detection method compared to four reference methods.

The next experiment aims at evaluating the binary change map generated from the change measure of each detector. To do that, a thresholding stage is carried out. In this study, we would like to investigate several techniques and observe the reaction of each detector for a general evaluation. Many thresholding methods have been studied in the literature. Here, we propose to perform three approaches including the *K-means* clustering [47], the *Otsu* thresholding [82] and the *Kittler Illingworth (KI)* thresholding [83]. For more details about these approaches, readers are invited to consult the mentioned papers. Basically, these thresholding methods are employed to automatically determine a threshold value from which the change measure is classified into two classes: *changed* and *unchanged*. From Figure 5.4(k) to 5.4(p), the binary change maps of the proposed detector and the four reference ones are shown. They are generated using the

K-means thresholding technique. We can observe that the results yielded by four reference methods (Figure 5.4(k) to 5.4(n)) seem to suffer the influence of speckle since they consist of many false alarms located outside the changed area. On the contrary, the result produced by the proposed method (Figure 5.4(p)) is more homogeneous and contains a very few number of outlier false alarms.

In order to perform a quantitative comparison, we propose to compute the ratio P_{GD}/P_{FA} and the percentages P_{TE} , P_{OA} obtained from each detector by using the three thresholding techniques. The experimental results can be found in Table 5.1. From this table, it is worth noting that there is always a compromise between the number of false alarms (FA) and missed detections (MD). If a detector yields a small number of false alarms, it may probably ignore the changed points and hence, increases the number of missed detections. As observed in the table, the *K-means* approach performs slightly better than the other two for most detectors. However, during our experiments, we observe that the threshold values calculated by the *K-means*, *Otsu* and *KI* approaches are quite close. Hence, despite a slightly better efficiency from *K-means*, any of the three can be selected for this thresholding stage. In terms of detection accuracy, the proposed method provides the best performance (highest P_{GD}/P_{FA} and P_{OA} , lowest P_{TE}) for all the three thresholding techniques. This issue confirms not only the effectiveness of the proposed strategy but also its robustness to different thresholding approaches.

Table 5.1 – Comparison of False Alarm rate (P_{FA}), Good Detection rate (P_{GD}), percentage of Total Errors (P_{TE}) and percentage of Overall Accuracy (P_{OA}) performed by five different change detection methods and three different thresholding techniques. Experiments are effectuated on data set 1 ($N_c = 2011$ changed points, $N_u = 7534$ unchanged points).

Thresholding method	Detector	FA (points)	MD (points)	GD (points)	P_{GD}/P_{FA}	P_{TE} (%)	P_{OA} (%)
<i>K-means</i>	MRD	804	690	1321	6.1555	15.65	84.35
	LRD	686	775	1236	6.7501	15.31	84.69
	GKLD	662	886	1125	6.3666	16.22	83.78
	CKLD	533	959	1052	7.3944	15.63	84.37
	Proposed	549	795	1216	8.2980	14.08	85.92
<i>Otsu</i>	MRD	840	652	1359	6.0611	15.63	84.37
	LRD	701	768	1243	6.6430	15.39	84.61
	GKLD	733	835	1176	6.0106	16.43	83.57
	CKLD	554	939	1072	7.2493	15.64	84.36
	Proposed	583	766	1245	8.0004	14.13	85.87
<i>KI</i>	MRD	723	754	1257	6.5134	15.47	84.53
	LRD	688	775	1236	6.7304	15.33	84.67
	GKLD	534	979	1032	7.2402	15.85	84.15
	CKLD	766	739	1272	6.2212	15.77	84.23
	Proposed	645	722	1289	7.4870	14.32	85.68

5.3.2.2 Results for data set 2

Figure 5.6 shows the two input ROIs of size 1000×850 pixels extracted from data set 2 and change detection results generated by the proposed method compared to the three reference ones including the MRD, the LRD and the GKLD. Here, change measure from each detector is thresholded by using the *Otsu* technique. We note that from our experiments, thresholding step performed by the *K-means* or the *KI* approaches also provides similar result to the *Otsu* technique.

Since the ground truth change mask for this data set is not available, only visual assessment can be considered. To help readers to have some ideas about where changes may occur between two input ROIs, Figure 5.6(c) shows the gray-scale change map obtained by the LRD (not the ground truth) in which change levels increase in function of intensity. As mentioned, these SAR data were acquired from an agricultural field with different periods of crop growth. Thematic changes between these images thus mainly come from these various stages of crops. Therefore, our expectation is to obtain a change map in which changed portions appear quite homogeneous. Moreover, the expected change detector needs to be robust to speckle.

According to Figure 5.6, the first observation is that the MRD (Figure 5.6(e)) yields a lot of over-detected points in its binary change map. On the contrary, the GKLD (Figure 5.6(g)) produces a great amount of missed detections since several changed portions remain undetected. Next, the LRD and the proposed detector (Figures 5.6(f) and 5.6(h), respectively) provide better performance. However, compared to our approach, the LRD still produces more false alarms. In order to clarify these remarks, let us consider and zoom in two studied crops from the top left and the bottom right of the ROI as shown in Figure 5.6(a). For a better explanation, these two image crops are named *Red* and *Green*. Change detection results for the *Red* crop are shown in Figure 5.7. Similarly, results for the *Green* crop can be found from Figure 5.8. For both cases, the proposed algorithm provides the best detection performance. By paying attention to the yellow circle from the *Red* crop, we consider a small detail which should not be detected as changed in our point of view. As observed, the MRD and the LRD detect it as changed (Figures 5.7(c)-(d)). Although the GKLD marks it as unchanged, this detector misses many other changed points (Figure 5.7(e)). On the contrary, the proposed method is capable of recognizing that unchanged detail as well as correctly providing the other changed portions (Figure 5.7(f)). Next, for the *Green* crop, we again observe the best performance from the proposed approach. The shape of the changed portion in Figure 5.8(f) is quite homogeneous and squared, which is relevant for scene interpretation from the data. Three reference detectors (Figures 5.8(c)-(d)-(e)) produce inferior results with many outlier false alarms. Consequently, the effectiveness of the proposed method can be confirmed and validated for data set 2.

5.3.3 Sensitivity to parameters

This subsection aims at analyzing the sensitivity of the proposed method to its parameters: the window size ω used for extracting local max keypoints, the number of closest neighbors K within graph construction stage and the patch size w_p considered for each keypoint. We propose to exploit the ROC plots performed on data set 1 for the experimental study.

5.3.3.1 Sensitivity to the density of keypoints (parameter ω)

This aspect involves the capacity of local maximum and minimum keypoints to represent the information carried by the image. In fact, in terms of visualization, one may observe some information loss from the image regions where there is no detected keypoint. However, thanks to

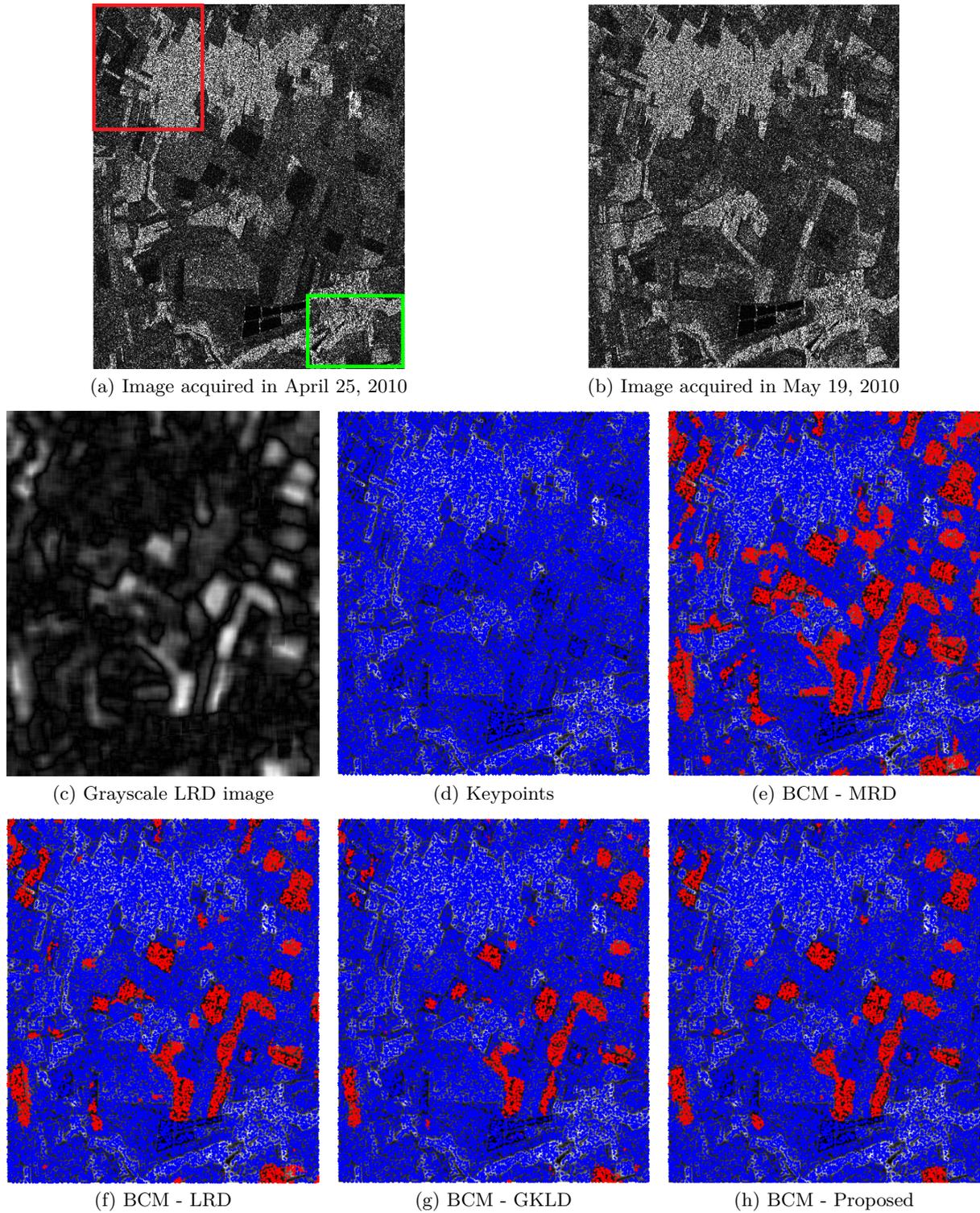


Figure 5.6 – Experimental study for Data set 2. (a) ROI from the image acquired in April 25. (b) ROI from the image acquired in May 19. (c) The grayscale LRD image. (d) Extracted keypoints. From (e) to (h) Binary change maps (BCM) yielded by different methods and thresholded using the *Otsu* approach. The two rectangles in red and green in (a) mark the positions of two studied crops (namely *Red* and *Green*) that will be analyzed in details in Figures 5.7 and 5.8.

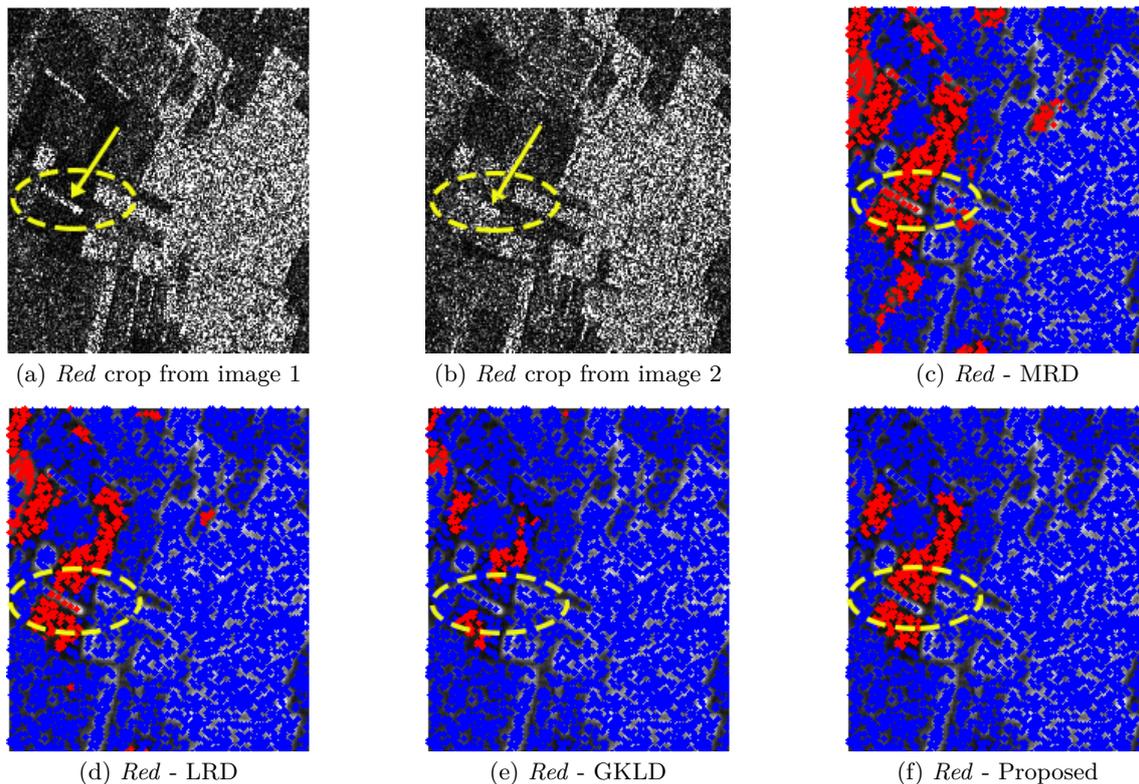


Figure 5.7 – Zoomed results for the *Red* studied crop of Data set 2 (see Figure 5.6(a)). (a)-(b) *Red* crop from 2 images. From (c) to (f) Corresponding binary change maps yielded by 4 detection methods for the *Red* crop, the yellow dash circle involves an interesting detail for discussion.

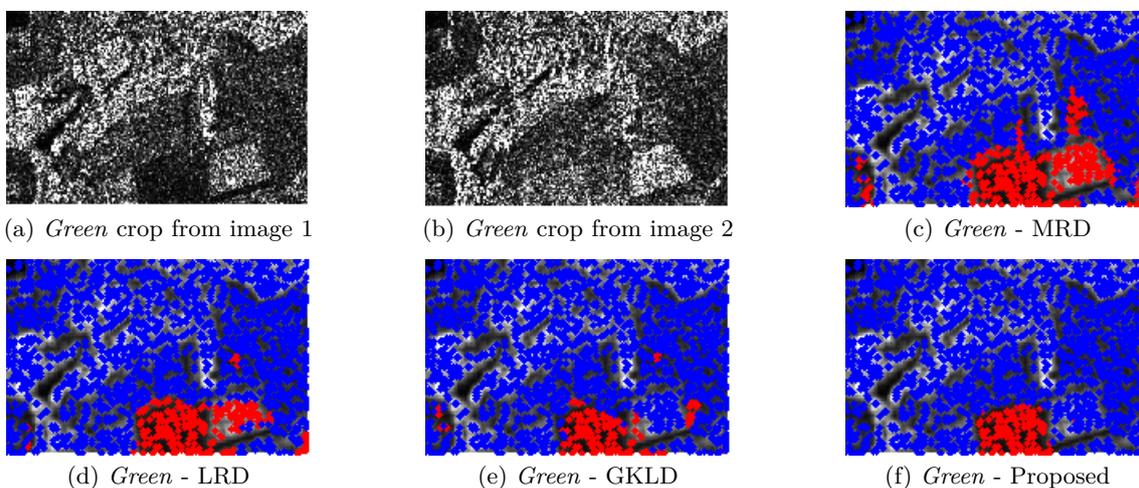


Figure 5.8 – Zoomed results for the *Green* studied crop of Data set 2 (see Figure 5.7(a)). (a)-(b) *Green* crop from 2 images. From (c) to (f) Corresponding binary change maps yielded by 4 detection methods for the *Green* crop.

their ability to capture most of the significant information of the image content (i.e. appearance within any intensity variation from the image), experiments show that these local extrema can produce an excellent image representation and approximation.

In Figure 5.9, ROC plots and the detection accuracy (P_{OA}) are used to quantitatively compare the performance of dense approach and pointwise approach (with different densities of keypoints) for change detection. We propose to perform the experiments employing the LRD since it can have both dense and pointwise versions. The *Otsu* technique is used for thresholding stage. We note that the density of local max keypoints is characterized by the window size (ω) used for keypoint extraction (Section 5.2.2). As observed from the figure, both dense and pointwise approaches (with ω varies from 3 to 7) provide very close performance in terms of ROC as well as P_{OA} . Then, in terms of complexity, the computational time of the pointwise approaches is significantly reduced compared to the dense one. This is because the dense approach detects changes for all pixels (image size = 800×400 , hence 320000 pixels), while pointwise approaches detect changes only on local max keypoints (9545, 3902 and 2974 points corresponding to the 3×3 , 5×5 and 7×7 window, respectively). Therefore, we can conclude that the sensitivity of the proposed algorithm to parameter ω is very weak. Moreover, if we ensure a good density of keypoints (i.e. ω is set from 3 to 9), such a pointwise approach can provide a stable detection performance without any concern about information loss.

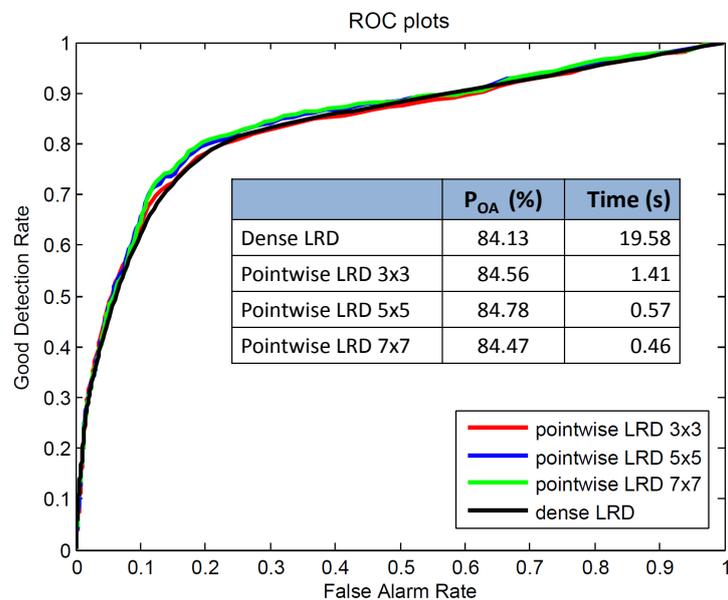


Figure 5.9 – Comparison of the dense and pointwise LRD methods (with different window sizes (ω) for keypoint extraction) for change detection.

5.3.3.2 Sensitivity to the number of closest neighbors (K) and the patch size (w_p)

The next experiment studies the parameter K . The patch size w_p is fixed to 15×15 pixels while K varies from 10 to 90. The corresponding result is shown in Figure 5.10(a). As observed, when K increases, the performance is considerably enhanced at the beginning (K from 10 to 50) but the enhancement slows down when K is greater than 70. Then, when K attains a value of 90 or higher, we observe a saturated phenomenon of ROCs. This behavior can be explained as follows. We have already mentioned that the parameter K implicitly represents the size of a neighborhood used for searching closest keypoints during graph construction stage. That is to say, K plays a similar role as a sliding neighborhood window considered for SAR image analysis and processing, due to the presence of speckle. It is proved that the more the

neighborhood is expanded, the better an operator can perform on SAR images, as long as SAR signal is considered to be stationary or at least locally stationary in the neighborhood. However, the performance will be saturated when the neighborhood size is too large, which explains this analogous behavior of K .

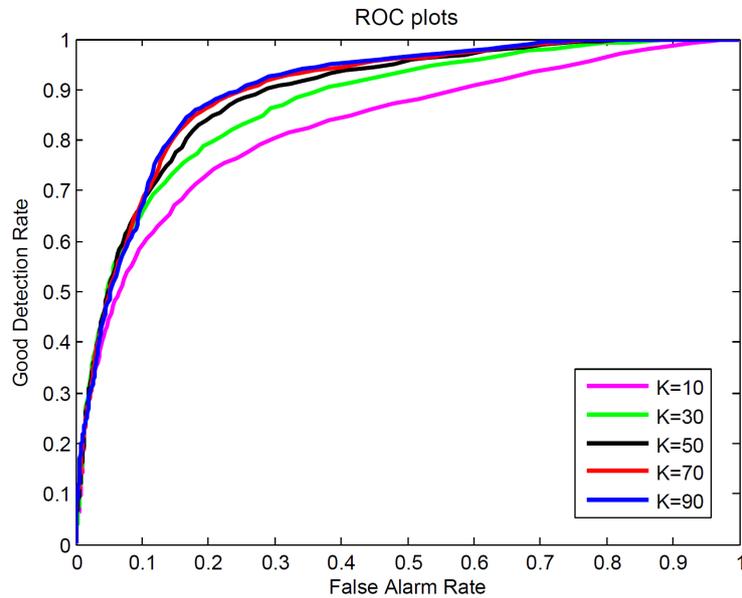
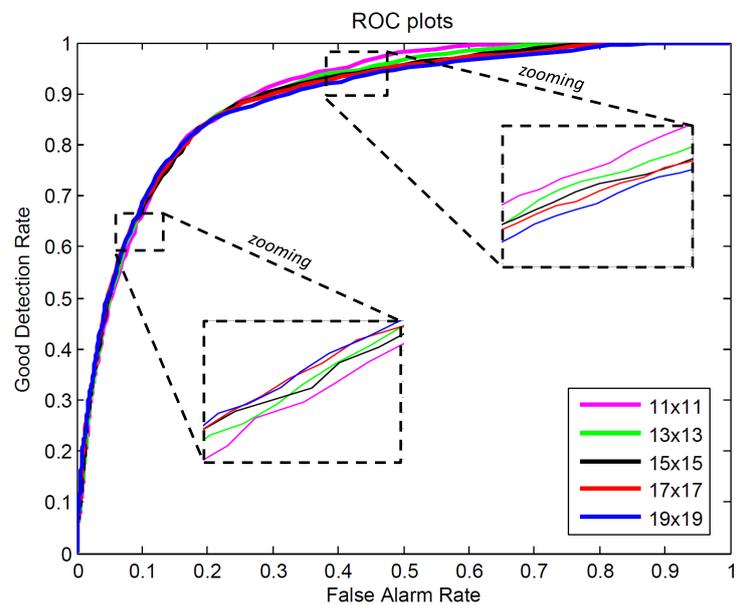
(a) Sensitivity to the number of closest neighbors K (b) Sensitivity to the patch sizes w_p

Figure 5.10 – ROC plots for the proposed method within the variation of its parameters: (a) The number of closest neighbors (K) for graph construction stage. (b) The patch size w_p considered for each keypoint.

Finally, the patch size w_p is studied within its variation from 11×11 pixels to 19×19 pixels. The parameter K is set to 50. The associated experimental result can be found in Figure 5.10(b).

We can observe that w_p does not create a strong impact on the algorithm's performance. Let zoom in two domains of the ROCs as done in the figure. It can be seen that large windows perform better within the first domain but less efficient during the second one. In addition, the variation of the five ROCs is not too high. These remarks enable us to conclude that the proposed method does not suffer an important influence from this parameter. We remind that only the first order statistic (i.e. the mean value) of the $w_p \times w_p$ pixel patch around each keypoint is required within our algorithm. Hence, a small patch size could be enough for the estimation. In consequence, although the proposed strategy is significantly dependent on the parameter K , the influence caused by w_p is quite inconsiderable.

5.3.4 Comparison with the NLM model

As remarked in the end of Section 5.2.4, in this subsection, we would like to distinguish the proposed method from the non-local mean (NLM) model proposed in [79]. We propose this comparison since readers may expect to do so when they reach Equation (5.8) in Section 5.2.4 which involves the generation of our change measure using the graph random walk matrix \mathcal{P} , instead of the graph adjacency matrix \mathcal{W} . For a reminder, using \mathcal{P} provides a change measure involving a normalized weighted summing of log-ratio distances derived from neighboring keypoints via graph structure (see Equation (5.8)), which has certain similarity to the NLM model.

The first important point to discriminate the two methodologies is the way each of them approaches and tackles the problem of change detection for SAR images. In fact, the NLM model involves a filtering process to reduce speckle noise (based on NLM denoising filter [84]) during change detection algorithm. Meanwhile, the proposed strategy aims at investigating the coherence of local structure and radiometric information between the two images. Our starting point is not a despeckling approach but texture tracking approach based on a pointwise image representation strategy via graph model. Let us summarize the two change measures using the NLM model proposed by the authors of paper [79]:

$$\text{CM}_1^{\text{NLM}}(p_n) = \text{LR} \left\{ I_1^{\text{NLM}}(p_n), I_2^{\text{NLM}}(p_n) \right\}, \quad (5.9)$$

$$\text{CM}_2^{\text{NLM}}(p_n) = \frac{1}{\sum_{p_k \in \mathcal{N}(p_n)} w_{\text{LR}}(p_n, p_k)} \sum_{p_k \in \mathcal{N}(p_n)} w_{\text{LR}}(p_n, p_k) I_{\text{LR}}(p_k), \quad (5.10)$$

where

- LR denotes the classical log-ratio operator,
- I_i^{NLM} , $i = \{1, 2\}$ is NLM filtered version of image i :

$$I_i^{\text{NLM}}(p_n) = \frac{1}{\sum_{p_k \in \mathcal{N}(p_n)} w_i(p_n, p_k)} \sum_{p_k \in \mathcal{N}(p_n)} w_i(p_n, p_k) I_i(p_k), \quad (5.11)$$

- I_{LR} is the log-ratio difference image derived from the two input images,
- $w_i(p_n, p_k)$, $i = \{1, 2\}$ and $w_{\text{LR}}(p_n, p_k)$ refer to the similarity weight between two pixels p_n , p_k computed from the image I_i , $i = \{1, 2\}$ and from I_{LR} , respectively.
- $\mathcal{N}(p_n)$ represents the dense search neighborhood around pixel p_n considered for NLM model [84].

Let us compare the two NLM change measures generated by Equations (5.9) and (5.10) to the measure yielded by our proposed method in Equation (5.8). A crucial remark is that the weighting function in NLM approach is calculated for each image (i.e. the first model, Equation (5.9)) or directly from the LR difference image (i.e. the second model, Equation (5.10)). On the contrary, in our strategy, similarity weight is computed only for the first image I_1 within graph construction stage.

Another significant point which makes our algorithm distinct from the NLM model is that for each point p_n , only a number of K closest neighboring keypoints are taken into account for change measuring. Meanwhile in NLM model, all pixels inside the dense search neighborhood $\mathcal{N}(p_n)$ are considered. This issue leads to the main drawback of NLM model, the computational complexity. If $W \times W$ refers to the size of $\mathcal{N}(p_n)$, we need to investigate all W^2 pixels for change measurement at p_n using NLM model. This amount is much higher than the number of points K within our method. Thus, using the proposed approach could reduce the complexity by the factor of K/W^2 in comparison with the NLM model. Moreover, it should be noted that our algorithm works based on keypoints which are previously selected. Thus, similarity weight functions can be directly derived from these points. On the contrary, the NLM model works based on dense neighborhood from which we need to calculate weights for every pixel inside that neighborhood and then select the most relevant ones. Furthermore, when K is fixed, the proposed method implicitly considers a search neighborhood of flexible size which varies depending on the density of keypoints around each keypoint p_n . For the NLM model, the size of $\mathcal{N}(p_n)$ is usually fixed, which ignores the proper local structure of the image at p_n . We will observe all of these remarks in more details during the analysis of the algorithm complexity of the proposed approach compared to reference methods in the next subsection.

5.3.5 Analysis of the algorithm complexity

Figure 5.11 shows an outline of different types of support neighborhood employed by different methods in order to measure the change level at each pixel. For a reminder, classical methods based on dense neighborhoods such as MRD, LRD, GKLD and CKLD use all pixels inside a fixed large-size window (denoted by $W \times W$, see Figure 5.11(a)) around the understudied pixel (marked in red). Next, the NLM and our method perform a weighted averaging operator of small patches within the larger neighborhood around the pixel to generate its change measure. However, as previously mentioned, while the NLM usually considers all patches inside the fixed-size $W \times W$ neighborhood (see Figure 5.11(b)), our method only takes into account patches at pre-selected keypoints within a flexible neighborhood whose size depends on the image's local properties around that pixel (see Figure 5.11(c)). These analyses reveal some ideas of the complexity of each method. In general, the computational time increases as follows: *classical dense methods (MRD, LRD, GKLD, CKLD) < proposed method < NLM method*.

Table 5.2 provides the complexity and calculation time of different methods together with their detection performance. Denote $w_p \times w_p$ the size of small patches within the NLM model and the proposed algorithm, the algorithm complexity can be estimated as $O(W^2)$ for classical dense methods (MRD, LRD, GKLD, CKLD), $O(W^2 w_p^2)$ for NLM model and $O(K w_p^2)$ for our method. Then from the third column of the table, we observe the effect of these complexity levels on their experimental computation time. A remark is that compared to NLM model (425.74s), the proposed strategy (15.94s) approximately reduces the complexity by a factor of K/W^2 (i.e. K is set to 50 and W is set to 35 in the experiment). Here, all the implementations are effectuated using MATLAB on a machine *Xeon 3.6GHz, 16 GB RAM*.

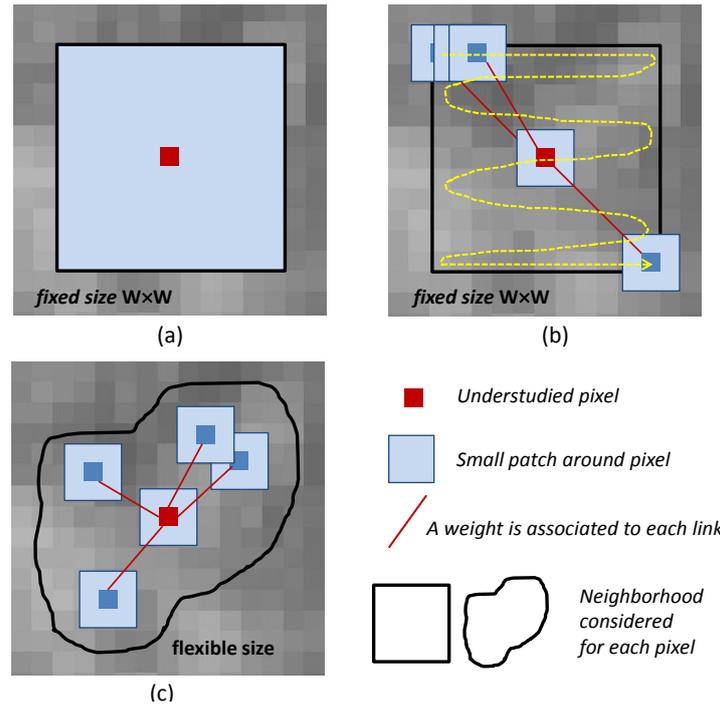


Figure 5.11 – Support neighborhood considered for measuring the change level at each pixel. (a) By MRD, LRD, GKLD and CKLD methods. (b) By NLM approach. (c) By the proposed approach.

Last but not least, one may wonder about the superior performance in terms of detection accuracy of the proposed approach compared to the NLM model (85.87% compared to 84.79%). In fact, our strategy only considers patches at keypoint locations for the averaging procedure. When choosing the local maximum pixels as keypoints, we are likely to select “good” patches for each central patch since the local maxima implicitly involve the notion of similarity among their patches. On the other hand, the NLM model accumulates all patches for its weighted averaging. In our experiment, the NLM approach has been implemented following Equation (5.10) without any optimization for weighting function or patch regularization because our main objective here is to study its complexity rather than to optimize its detection performance. More details about the optimization of the NLM model can be found in the related paper [79]. In conclusion, the efficiency of the proposed method consists of a superior detection performance and an intermediate complexity between classical dense methods and the NLM approach.

Table 5.2 – Comparison of computational complexity and detection performance from different methods. Experiments are performed on data set 1 using the *Otsu* thresholding.

Method	Complexity	Computation time (s)	P_{GD}/P_{FA}	POA (%)
MRD	$O(W^2)$	1.51	6.0611	84.37
LRD	$O(W^2)$	1.54	6.6430	84.61
GKLD	$O(W^2)$	2.08	6.0106	83.57
NLM	$O(W^2 w_p^2)$	425.74	7.0186	84.79
Proposed	$O(K w_p^2)$	15.94	8.0004	85.87

5.3.6 Exploitation of the local minimum pixels

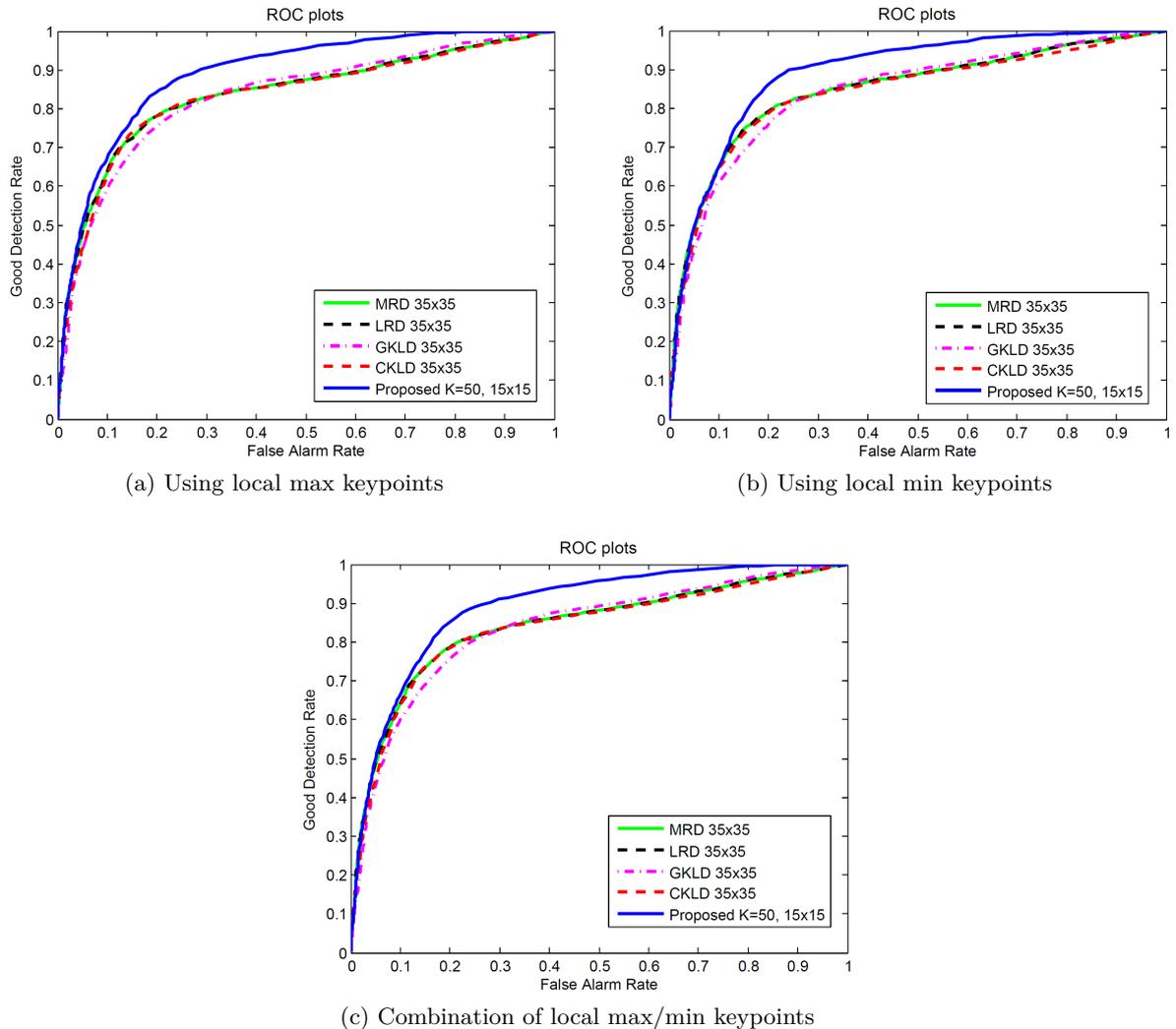


Figure 5.12 – ROC plots for the proposed change detection method compared to four reference methods in case of using: (a) the local max keypoints; (b) the local min keypoints; (c) the combination of both local max/min keypoints to perform the pointwise strategy.

As discussed in Section 5.2.2, the local minimum pixels can be used to replace the local maxima for graph construction stage. Figure 5.12 confirms similar performance of the algorithm when using the local max keypoints, the local min keypoints or both of them to perform change detection task. The experiment is conducted using data set 1. In this case, a slightly better performance is achieved by using the local max keypoints (Figure 5.12(a)) due to the nature of changes occurring within the image. Nevertheless, using the local min keypoints or the combination of both local extrema could also provide a good detection performance compared to the other reference methods.

5.4 Application to glacier displacement detection

In this section, the proposed texture tracking technique is applied to the measurement of glacier flows over time using SAR images. The objective is to search for the displacement vector which shows the flow magnitude and direction of each interest pixel located on the glacier. We continue to exploit the dissimilarity measure generated based on the weighted graph structure to search for the most relevant displacement of each keypoint between two SAR images. It is worth noting that this application is not a main task of the chapter. It is proposed to check the behavior of graph-based texture tracking strategy to detect displacements instead of measuring changes between two images. Let us now describe the proposed method and provide some preliminary efforts experimented on TerraSAR-X data.

5.4.1 Proposed method

The proposed method for glacier displacement detection is highlighted in Figure 5.13. Let us consider the two SAR images namely the Master and the Slave for this study. In general, the Master image is the one which is chronologically acquired before and the Slave is captured later. Similar to our change detection framework (Figure 5.1), a set of local max keypoints are first extracted from the Master. A weighted graph is then constructed to connect all keypoints as described in Section 5.2.3. For each understudied keypoint from the Master, we search from the Slave the corresponding pixel which is the most adapted to the local graph structure located at the keypoint. In other words, the local texture information around each keypoint is tracked with the help of the constructed graph structure. In practice, the searching process for each keypoint is limited within a research zone, not from the whole image, due to the prior knowledge of the maximum displacement occurring from the two images.

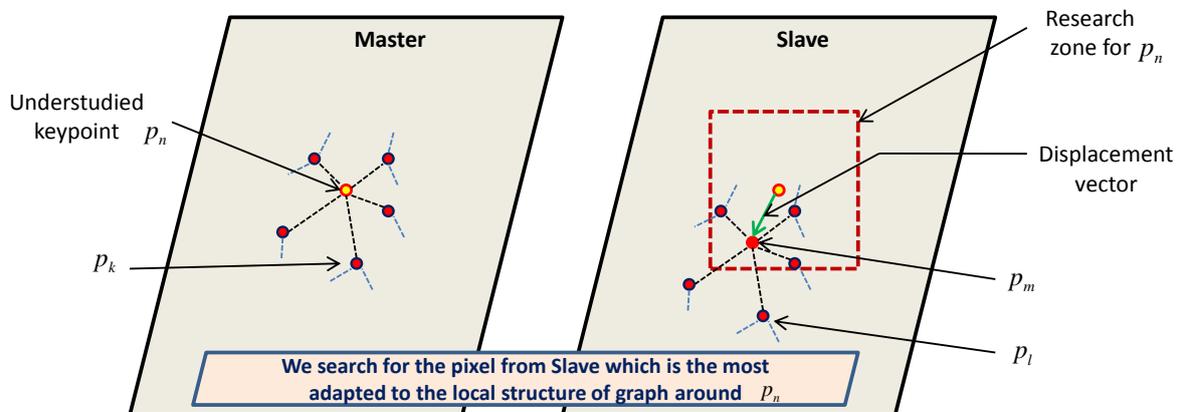


Figure 5.13 – Outline of the proposed graph-based texture tracking for glacier flow detection.

Denote p_n the keypoint located at position (x_n, y_n) on the Master plane having the intensity $I^M(p_n)$. The displacement vector at p_n is calculated:

$$\vec{v}(p_n) = p_m^* - p_n, \quad (5.12)$$

$$p_m^* = \arg \min_{p_m \in \mathcal{Z}(p_n)} D(p_n, p_m), \quad (5.13)$$

where:

- p_m : pixel located at position (x_m, y_m) on the Slave.

- p_m^* : corresponding pixel of p_n which minimizes the graph-based dissimilarity measure.
- $\mathcal{Z}(p_n)$: the research zone to search for the corresponding pixel of p_n .
- $D(p_n, p_m)$: the dissimilarity measure (i.e. distance measure) between p_n from the Master and p_m from the Slave generated with the help of the local graph structure at p_n . It refers to how much the local information around p_m still adapts to the graph structure at p_n . It is computed as the change measure defined in Equation (5.7):

$$D(p_n, p_m) = \sum_{p_k \sim p_n} w(p_k, p_n) \left| \log \frac{I^M(p_k)}{I^S(p_l)} \right|, \quad (5.14)$$

in which $p_l = (p_m - p_n) + p_k$ (the connection between p_m and p_l on the Slave corresponds to the connection between p_n and p_k on the Master). The weight function is generated from the Master image:

$$w(p_k, p_n) = \exp \left\{ - \left| \log \frac{I^M(p_n)}{I^M(p_k)} \right| \right\}. \quad (5.15)$$

5.4.2 Preliminary results

Experiments are performed on the two SAR intensity images acquired from the Argentiere Glacier located in the French Alps. These images were captured by the TerraSAR-X (TSX) satellite using the Stripmap descending mode with the spatial resolution of 1.4m in range and 2.5m in azimuth. Here, we study the glacier displacement from the two images acquired in 12/7/2009 and 14/8/2009. They are shown in Figures 5.14 together with an optical image illustration at the top.

Figure 5.15 shows the glacier displacement result provided by the proposed strategy. Here, the red vectors indicate displacements in good directions (the glacier flow direction is known). The green ones are in bad directions. The experiment is carried out by setting a search window size 5×5 pixels ($\omega = 5$) for the local max keypoint extraction. Then, the weighted graph is built by preserving $K = 200$ connections for each vertex. We note that both images are smoothed by using a 3×3 Gaussian low-pass filter to reduce the effect of speckle noise.

From Figure 5.15(a), about 94% of displacements are detected in good direction (marked in red). Let us analyze the result by zooming in two regions of interest (ROI) marked by the two rectangles. For better explication, they are named *Yellow* crop and *White* crop. The *White* crop is extracted from a homogeneous (i.e. smooth) region of the glacier. The *Yellow* one is cut from a crevasse area with rough surface and very structural texture produced by fast-moving glaciers down to the serac fall. The zoomed results of these two crops are shown in Figures 5.15(b) and (c), respectively. The result of the *White* crop consists of several false detections (green vectors). Also, red vectors seem to be quite different in direction and magnitude. Hence, the performance of the proposed algorithm is limited for this crop. On the other hand, the detection result for the *Yellow* crop at crevasse region is consistent. All detected vectors reveal the glacier flow direction. Then, these vectors' length tends to increase at the serac fall. The maximum vector flow attains 9 pixels in range and 4 pixels in azimuth which corresponds to a displacement of about 16m in 33 days (duration between two acquisition dates). Hence, the maximum displacement at serac fall is estimated to be 0.48m/day. We note that the annual measurement estimates that the displacement at this serac fall region is about 0.4m/day [85].

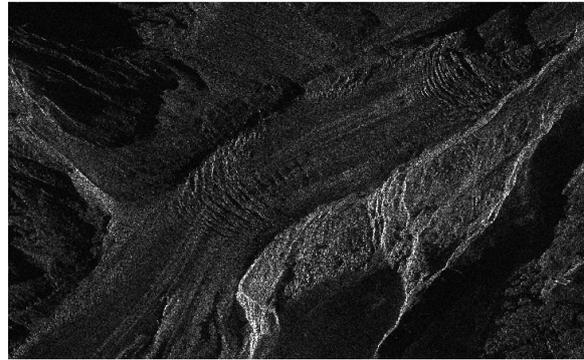
The above observations show that our strategy is able to provide good performance on crevasse areas with strong and structural texture information. However, the performance is limited for homogeneous and smooth regions. In fact, glacier flow detection using texture coherence



(a) Argentiere glacier



(b) TSX image acquired in 12/07/2009



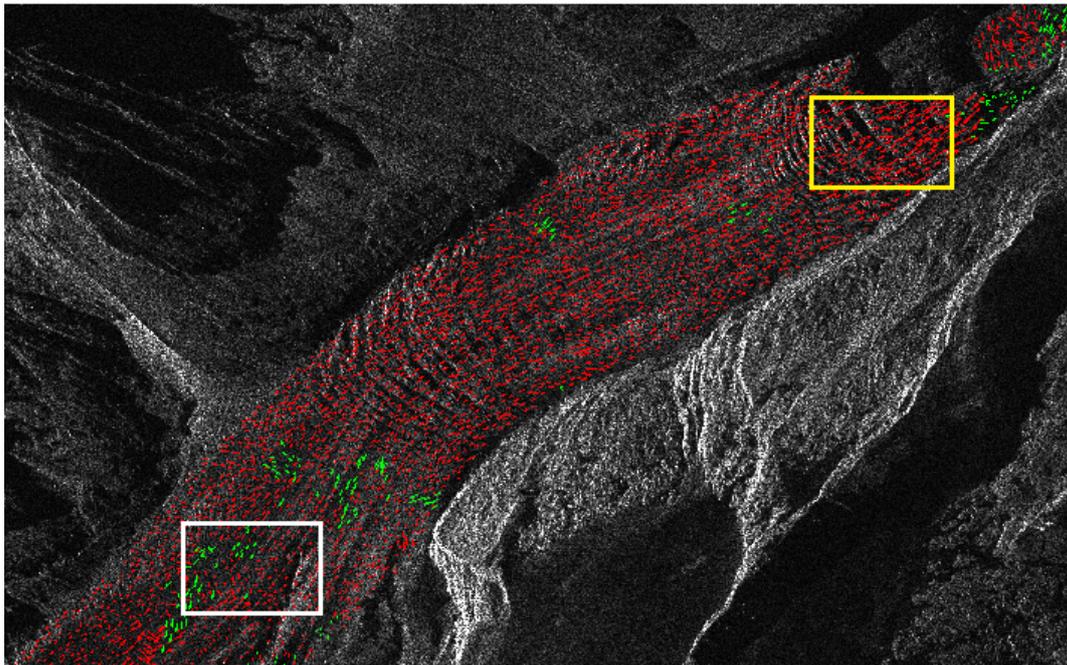
(c) TSX image acquired in 14/08/2009

Figure 5.14 – Input TerraSAR-X images acquired from the Argentiere Glacier, ©Airbus DS.

and texture tracking finds it challenging within homogeneous and flat areas. This behavior is also found in previous studies [85, 86]. Our method works based on the coherence of texture information encoded by a graph structure. The dissimilarity measure in Equation (5.14) is not robust enough to deal with flat areas like the *White* crop. More actions are required to model those homogeneous textures. However in general, the proposed technique can provide a fast estimation of displacements, especially for crevasse surface of the glacier.

5.5 Conclusion

This chapter has presented the second application of the proposed pointwise approach in combination with signal processing on graphs. A pointwise texture tracking technique has been developed to tackle change detection task using SAR images. Unlike the previous chapter where textural features are described based on the spectral domain of weighted graph, we exploit here graph characteristics directly from the vertex domain to encode texture information. Change measure has been derived from the coherence and compatibility of intensity information carried by the two images with the support of the constructed graph's local structure. Experimental results have demonstrated the superior performance of the proposed algorithm compared to reference methods, in terms of detection accuracy and robust behavior to different thresholding techniques. In addition, some efforts have been made to apply the proposed texture tracking technique to the detection of glacier displacement. Our method seems to have a good behav-



(a) Glacier displacement result

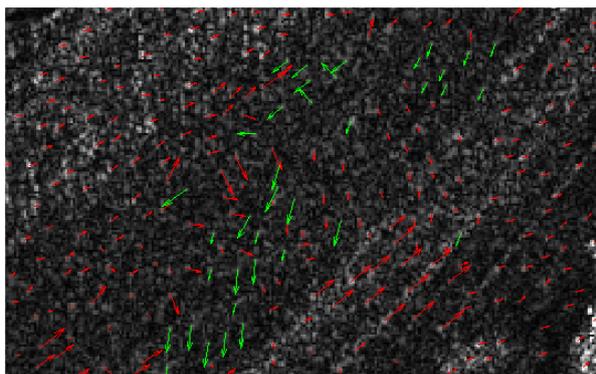
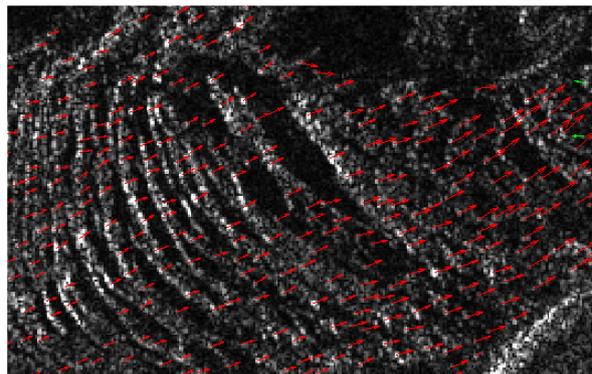
(b) Zoomed result from *White* crop(c) Zoomed result from *Yellow* crop

Figure 5.15 – Glacier displacement result yielded by the proposed algorithm. Vector flows with good direction are marked in red.

ior on rough surfaces located on crevasse regions, but not on homogeneous areas. Hence, more studies need to be devoted to improve the performance for this thematic application. We discuss this issue in Chapter 8.

We now close the combination of pointwise approach and graph theory. In the next part, we refocus on developing effective methods for texture description from VHR optical images. Our motivation is to perform the pointwise approach with taking into account structural features from the image content in order to produce efficient descriptors in terms of good texture discrimination as well as low algorithm complexity.

Part III

Pointwise approach combined with structural features

In this part, we propose two novel local descriptors to encode textural information using the proposed pointwise approach. The main objective of this part is to take into account structural features which have not been exploited in previous chapters. We will not continue using graph theory and the reason will be explained during the next two chapters. In order to encode textural and structural features, Chapter 6 first studies their correlation and joint distribution inside a local neighborhood by employing a covariance-based technique. Then, Chapter 7 develops the local extrema descriptor and proves its relevance for texture retrieval and classification tasks applied to optical remote sensing images.

PW-COG: A pointwise covariance descriptor for texture characterization

Everything is theoretically impossible, until it is done.

Robert A. Heinlein

6.1	Introduction	98
6.1.1	Motivation of the chapter	98
6.1.2	Chapter overview	99
6.2	Pointwise construction for covariance-based descriptors	99
6.2.1	Region covariance matrix of oriented gradients	99
6.2.2	Pointwise construction of COG	100
6.2.3	Adapted distance metric for covariance descriptors	101
6.3	Application to texture-based VHR image classification	102
6.4	Experimental study	103
6.4.1	Data description	103
6.4.2	Experiment setup	104
6.4.3	Results and discussion	107
6.4.4	Sensitivity to parameters	116
6.4.5	Sensitivity to the distance metric	117
6.5	Conclusion	118

The content of this chapter is mainly based on the following published papers:

- M. T. Pham, G. Mercier, J. Michel, *PW-COG: an effective texture descriptor for VHR satellite imagery using a pointwise approach on covariance matrix of oriented gradients*, IEEE Trans. Geosci. Remote Sens., vol 54, issue 6, pp. 3345-3359, 2016.
- M. T. Pham, G. Mercier, J. Michel, *Pointwise approach for covariance matrix of oriented gradients for very high resolution image texture segmentation*, Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Milan, Italy, 2015, pp. 1008-1011.

6.1 Introduction

6.1.1 Motivation of the chapter

As introduced, this chapter goes back to investigate the problem of texture description based on characteristic points in the scope of VHR optical imagery. For a reminder, in Chapter 4, we developed a pointwise graph-based method for texture characterization by integrating the non-dense approach based on the local maximum and local minimum pixels into a weighted graph model. Textural features were extracted using the spectral signatures of the constructed graph. The method was proved to be more efficient than some classical dense approaches when applied to unsupervised classification framework for VHR panchromatic and multispectral images. However, the main drawback of this work is the relatively high computational cost which is mainly resulted by the graph construction stage. In fact, during the construction of weighted graph to connect keypoints, all pairwise distances of vertex's feature vectors need to be calculated, which significantly increases the computation time (see Chapter 4, Section 4.4.4). That is why in this chapter, we would like to carry out a direct texture description algorithm for VHR images by exploiting such a pointwise approach without constructing any graph structure. This straightforward method is expected to considerably reduce the algorithm's complexity. Another problem of the proposed descriptors in Chapter 4 is their weak capacity to discriminate different oriented texture zones occurring within the image. The reason is that only radiometric and spatial information were encoded to generate the feature vectors serving for graph construction. Hence, the second objective of the present chapter is to take into account structural features of the image to improve the capacity of the new descriptor to deal with oriented textural features.

In [87], the region covariance matrix is proposed as an interesting and promising local feature descriptor for detection and recognition problems in the domain of computer vision and image, video processing. By employing covariance-based descriptors performed on the image gradient features, recent studies have tackled several object detection tasks such as visual feature tracking [88] as well as head pose estimation [89]. Considered as a natural way of fusing multiple features holding some correlated properties, such a feature covariance matrix makes it possible to take into account both the variance and the joint distribution of features. Furthermore, by integrating radiometric information and oriented gradient features which capture the image geometric and structural properties, this kind of descriptor is quite relevant for the main task of texture analysis and characterization in our work.

Therefore, the main motivation of this chapter is to construct the covariance matrix of oriented gradients (COG) using a pointwise approach based on characteristic points to characterize textural features for VHR optical images. That is to say, only radiometric and gradient features located at characteristic pixels will be considered during the COG construction. The novel descriptor is named PW-COG (i.e. pointwise covariance matrix of oriented gradients). We continue to exploit the local maximum and local minimum pixels as our favorite keypoints for the non-dense representation and description of different texture zones from VHR images. Along with the aim of reducing computational time, the proposed descriptor is expected to possess a strong capacity of distinguishing different oriented textures by using only panchromatic VHR image data. In order to assess the effectiveness of the proposed descriptor, both texture-based unsupervised and supervised image classification tasks are performed. We employ the *Brodatz* texture database and the VHR panchromatic Pléiades images to demonstrate and confirm its superior performance compared to reference methods, in terms of classification accuracy as well as algorithm complexity.

6.1.2 Chapter overview

We organize the remainder of this chapter as follows. Section 6.2 reviews the classical covariance-based descriptor and presents the pointwise approach for its construction. The proposed PW-COG descriptor is next integrated into a texture-based classification framework applied to VHR panchromatic images in Section 6.3. Then in Section 6.4, we describe our experimental study and provide both unsupervised and supervised classification results yielded by the proposed algorithm compared against some reference approaches, including the previous pointwise graph-based descriptors in Chapter 4. Qualitative and quantitative assessment will be delivered for a validation of the proposed novel descriptor. Section 6.5 finally concludes the chapter.

6.2 Pointwise construction for covariance-based descriptors

6.2.1 Region covariance matrix of oriented gradients

Image gradients are considered as important parameters for human perception and analysis of visual features including textures. They have so far become key features in several local descriptors such as the Scale-Invariant Feature Transform (SIFT) [25], the Gradient Location and Oriented Histogram (GLOH) [31], the Histograms of Oriented Gradients (HOG) [90] and the Histogram of Second Order Gradients (HSOG) [32], etc. These descriptors are basically generated by the concatenation of histograms formed by the quantization of gradient locations, magnitudes and orientations. They have been successfully applied to a variety of image and video processing tasks such as visual object detection and tracking, image retrieval, recognition and registration, etc. In [87], the region covariance of gradient features is proposed as a fast descriptor to tackle detection and classification problems. Unlike other local descriptors based on histogram computation, this covariance-based descriptor is constructed via a matrix model which is capable of encoding both the variance of each feature (i.e. matrix’s diagonal entries) as well as their correlations (i.e. matrix’s off-diagonal entries). Since these statistical characteristics take into account the correlated properties of different parameters including the radiometric information (i.e. intensity) and several gradient features, they help to enhance the texture distinctiveness capacity of the descriptor.

In order to construct the covariance descriptor, the feature image needs to be first extracted. In this work, it consists of the intensity, the first and the second-order gradient features. First, the image intensity is used to provide radiometric information which encodes the capacity of illumination reflection of objects and features within different texture zones. Hence, it becomes quite significant to human perception of visual textures. Second, oriented gradients are capable of capturing the image’s local geometry, which is appropriate for texture representation and discrimination. Here, both the first-order and second-order gradients are employed since they are proved to be able to provide complementary information from the image geometric properties [31,32]. More specifically, the first-order gradients describe the slope and the elasticity of surfaces that characterize the image local geometry. On the other hand, the second-order gradients involve the surfaces’ principal curvatures and local variations. In summary, the feature image is computed as follows:

$$F = \left\{ f_p \in \mathbb{R}^6 / f_p = [I_p, I_p^x, I_p^y, I_p^{xx}, I_p^{yy}, I_p^{xy}]^T \right\}, \quad (6.1)$$

where I_p^x (resp. I_p^y) stands for the horizontal (resp. vertical) gradient and I_p^{xx} , I_p^{yy} , I_p^{xy} denote three second-order gradients computed at pixel p from the image I . We note that if the size of I is $M \times N$ pixels, the size of F will be $M \times N \times 6$.

Let us recall the construction of the region covariance descriptor. Consider an image region (i.e. image neighborhood) of size $W \times W$ pixels around a studied pixel p , denoted by $\mathcal{N}_W(p)$ including W^2 pixels where each pixel q is characterized by a feature vector f_q consisting of both intensity and different gradient features. The COG matrix of region $\mathcal{N}_W(p)$ associated to the pixel p , denoted by C_p , can be generated as follows:

$$C_p = \frac{1}{W^2} \sum_{q \in \mathcal{N}_W(p)} (f_q - \mu_p)(f_q - \mu_p)^T, \quad (6.2)$$

where μ_p is the estimated mean feature vector of the neighborhood $\mathcal{N}_W(p)$:

$$\mu_p = \frac{1}{W^2} \sum_{q \in \mathcal{N}_W(p)} f_q. \quad (6.3)$$

It is worth noting that the noise perturbation in the image is significantly reduced by the averaging operator during the COG computation. The size of region (W) is an important parameter of the descriptor that indicates the scale at which we expect to characterize local features for each pixel, like most of other descriptors (e.g. local feature histogram, gray-level cooccurrence matrices, etc.).

6.2.2 Pointwise construction of COG

Covariance-based local descriptors are traditionally generated using a dense approach which takes into consideration all image pixels inside a neighborhood around each given pixel, as described above. Now, we propose to construct these descriptors using a pointwise strategy. It means that only feature vectors located at characteristic points are taken into consideration to generate the covariance matrix. In the scope of this thesis, characteristic points continue to be the local maximum and local minimum pixels thanks to their capacity to capture significant texture information of the image (cf. Chapter 2). To construct the covariance descriptor for an understudied pixel p , the first step is to search for its closest local maximum and closest local minimum pixels. We have mentioned two strategies which can be applied to this searching step in Section 4.2 (Chapter 4). For a recall, they are:

1. Fixing the number of K closest local maxima and K closest local minima around p .
2. Fixing a window size $W \times W$ around p . Then, all local maxima and minima inside that window are considered.

The advantages and disadvantages of each strategy were previously discussed. In Chapter 4, the first one was exploited to construct the pointwise (PW) descriptor served for graph construction. We also remarked that both strategies could provide similar performance of the generated descriptor. In this chapter, the second approach considering a fixed window size for all keypoints is employed. Another advantage of this approach is that its implementation is less costly than that of the first one, especially for large-size image data (i.e. the search of K closest points is more costly within large-size images). Nevertheless, both approaches can be employed.

Now, for each studied keypoint p , the pointwise COG descriptor (denoted by $\delta^{\text{PW-COG}}(p)$) including two covariance matrices is constructed as follows:

$$\delta^{\text{PW-COG}}(p) = \{C_p^{\max}, C_p^{\min}\}, \quad (6.4)$$

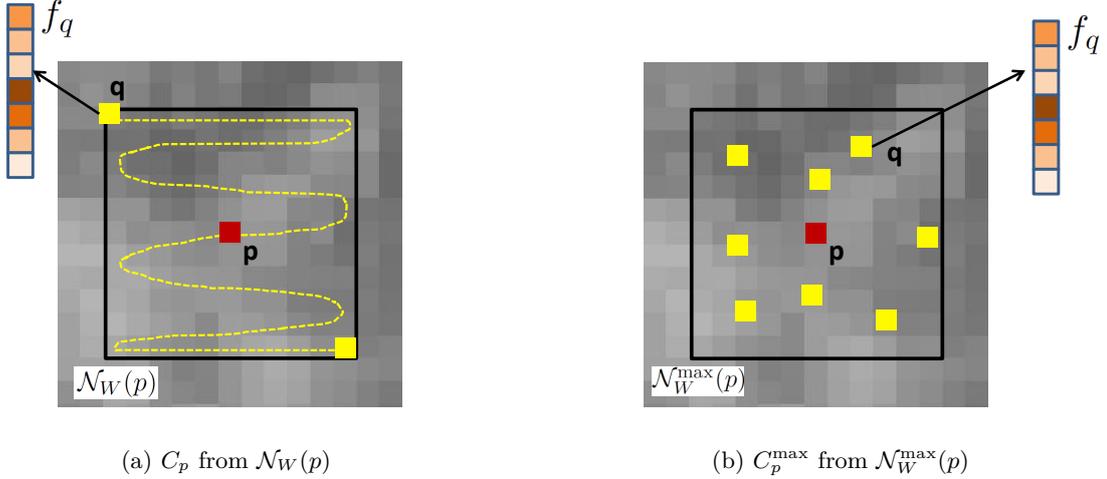


Figure 6.1 – Generation of dense COG from the neighborhood $\mathcal{N}_W(p)$ and pointwise COG from the local maximum set $\mathcal{N}_W^{\max}(p)$.

where

$$C_p^{\max} = \frac{1}{|\mathcal{N}_W^{\max}(p)|} \sum_{q \in \mathcal{N}_W^{\max}(p)} (f_q - \mu_p^{\max})(f_q - \mu_p^{\max})^T, \quad (6.5)$$

$$C_p^{\min} = \frac{1}{|\mathcal{N}_W^{\min}(p)|} \sum_{q \in \mathcal{N}_W^{\min}(p)} (f_q - \mu_p^{\min})(f_q - \mu_p^{\min})^T, \quad (6.6)$$

where μ_p^{\max} (resp. μ_p^{\min}) is the mean feature vector computed as in Equation (6.3) for the set $\mathcal{N}_W^{\max}(p)$ (resp. $\mathcal{N}_W^{\min}(p)$). $|\mathcal{N}|$ represents the cardinality of the set \mathcal{N} (i.e. the number of local maxima or local minima in the dedicated set).

Since these two matrices are symmetrical, it is sufficient to store only the entries from their upper (or lower) triangular part in $\delta^{\text{PW-COG}}(p)$. In our algorithm, the matrix size is 6×6 . Thus, the descriptor vector length should be $2 \times 21 = 42$.

Figure 6.1 summarizes the generation of COG from a dense neighborhood $\mathcal{N}_W(p)$ and from a non-dense neighborhood $\mathcal{N}_W^{\max}(p)$. In the first case (Figure 6.1(a)), all feature vectors f_q inside $\mathcal{N}_W(p)$ are considered to calculate C_p as Equation (6.2). The total number of feature vectors is W^2 . On the other hand, only f_q located at local maxima positions are taken into account to construct C_p^{\max} as Equation (6.5). In this case, the number of considered feature vectors is $|\mathcal{N}_W^{\max}(p)| \ll W^2$. We note that the same principle is applied to compute C_p^{\min} .

Last but not least, it should be noted that other radiometric and geometric features can be integrated into the computation of the above keypoint-based descriptor. For example, one can employ gradient parameters from other directions, others may expect to insert the magnitude and orientation values into their feature images, etc. Here, we construct our own descriptor using the image intensity together with its first and second-order derivatives regarding to the two basic horizontal and vertical directions. That is to say, the proposed strategy can be employed and adapted to other kinds of features and it remains an open issue of the fundamental feature selection problem depending on one's expectation to form their own descriptor.

6.2.3 Adapted distance metric for covariance descriptors

An important remark is that covariance-based descriptors do not lie on the Euclidean space. Hence, a dedicated distance metric is required for measuring their dissimilarity. In [91], an

adapted distance measure is proposed due to the fact that COG matrices possess a positive definite structure which conforms to the Riemannian manifold space. Hence, a geometric-based distance metric, called Riemannian metric, is considered rather than an arithmetic-based metric which is usually applied to Euclidean space. The following formula allows us to calculate the Riemannian distance between two covariance matrices:

$$\rho(C_1, C_2) = \sqrt{\sum_{\ell=1}^d \ln^2 \lambda_{\ell}}, \quad (6.7)$$

with

$$\lambda_{\ell} C_1 \chi_{\ell} - C_2 \chi_{\ell} = 0, \quad \ell = 1, \dots, d, \quad (6.8)$$

where $d \times d$ is the dimension of C_1 and C_2 ($d = 6$ in this chapter's work). λ_{ℓ} and χ_{ℓ} are the ℓ^{th} generalized eigenvalue and eigenvector which satisfy Equation (6.8).

Now, if we want to compute the Riemannian distance between PW-COG descriptors of two keypoints p_1 and p_2 , the following equation is applied:

$$\text{dist} [\delta^{\text{PW-COG}}(p_1), \delta^{\text{PW-COG}}(p_2)] = \rho(C_{p_1}^{\text{max}}, C_{p_2}^{\text{max}}) + \rho(C_{p_1}^{\text{min}}, C_{p_2}^{\text{min}}). \quad (6.9)$$

6.3 Application to texture-based VHR image classification

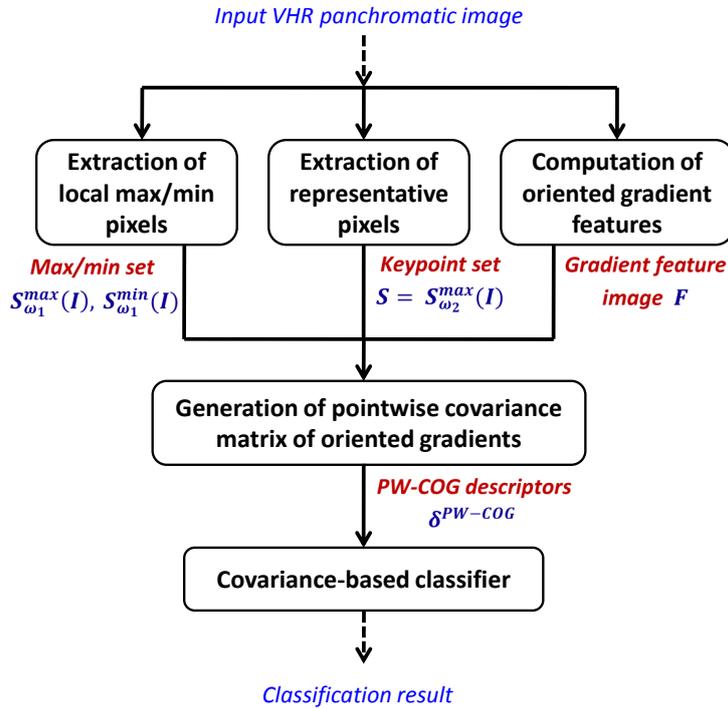


Figure 6.2 – Proposed framework for texture-based classification of VHR panchromatic images using the proposed PW-COG descriptor.

In this section, the complete classification framework using the proposed PW-COG descriptor is presented. An overview of the proposed strategy is highlighted in Figure 6.2. The first two blocks involve the extraction of the local maximum and local minimum pixels as well as the

extraction of representative pixels (which are also the local maximum keypoints as in Chapter 4) used for our keypoint-based approach. The reason why the local maximum pixels are also exploited as representative pixels was explained in Section 4.3.1 (Chapter 4). The next block on the right represents the generation of a feature image F including radiometric and gradient properties derived from the image. This feature image and all keypoint sets are then exploited to produce the PW-COG descriptors. Based on these descriptors, classification stage is finally carried out for the evaluation of the descriptor performance. In details, the full algorithm applied to an input VHR panchromatic image I is implemented as follows:

1. Extract the local maximum set $S_{\omega_1}^{\max}(I)$ and the local minimum set $S_{\omega_1}^{\min}(I)$ from the image using a search window $\omega_1 \times \omega_1$. These two sets will be exploited for the generation of pointwise covariance descriptors.
2. Extract the keypoint set S to represent and perform the keypoint-based classification algorithm. As in Chapter 4, we choose $S = S_{\omega_2}^{\max}(I)$ with $\omega_2 \geq \omega_1$ to speed-up the processing time.
3. Generate the feature image F consisting of the image intensity and gradient features (both first-order and second-order) as in Equation (6.1).
4. For each keypoint $p \in S$, search for its neighboring local max pixels from $S_{\omega_1}^{\max}$ and neighboring local min pixels from $S_{\omega_1}^{\min}$ inside the $W \times W$ window centered at it, denoted by $\mathcal{N}_W^{\max}(p)$ and $\mathcal{N}_W^{\min}(p)$. Then, extract $\delta^{\text{PW-COG}}(p)$ as in Equation (6.4).
5. For the final classification stage (i.e. unsupervised or supervised), exploit the adapted distance measure between two PW-COG descriptors defined in Equation (6.9).

The above algorithm can be applied to perform both unsupervised and supervised classification tasks. Two key features would be the generation of PW-COG descriptors and the adapted metric used for distance measurement. Another remark is that the proposed method involves three parameters: the window size used for the extraction of local max and local min pixels (ω_1), the window size for the extraction of representative max keypoints (ω_2) and the window size set for the generation of PW-COG descriptors (W). The sensitivity to each one will be discussed in details within our experimental study in the next section.

6.4 Experimental study

6.4.1 Data description

In this chapter, the main processed data are two images acquired by the VHR Pléiades satellite, copyright CNES©. They were acquired by the satellite's panchromatic sensor with a spatial resolution of 70cm at nadir, resampled at 50cm. Our implementation exploits the GEOTIFF 12-bit image data. The first image acquired in June 24, 2012 covers an intensively agricultural area of 80 km² near the Tagon-Marcheprime region located between Bordeaux and Arcachon, France. The scene consists of different types of cultivation including many parcels of vegetation, pine trees or vineyard areas, etc. They are planted in different row orientations with various densities. Besides grown crops, there exist other types of classes such as grass pasture, road and bare soil. All of these land-cover characteristics introduce a variety of texture zones in the image acquired from the scene. The second image acquired in April 25, 2013 covers a 180 km² area around the Arcachon Bay, France. In addition to rural regions including different cultivation

types and forest zones like in the first image, this image also contains several urban areas which involve more complicated textural and structural features.

Two regions of interest (ROI) are extracted from the first image to perform our unsupervised classification experiments. The first ROI of 900×1000 pixels, namely *ROI-1*, shown in Figure 6.4(a) consists of five texture types including the pine forest, two oriented early-grown vine parcels, roads and bare soil. In the second ROI of 800×1300 pixels, namely *ROI-2* (Figure 6.5(a)), we can find vine parcels with different row directions, pine tree forests, roads as well as grass pasture zones. Next, from the second image, another ROI, namely *ROI-3*, consisting of both rural and urban zones is extracted for supervised classification experimentation. This ROI has the size of 2000×3200 pixels and can be seen in Figure 6.6(a).

Besides the above VHR Pléiades images, we also propose to employ the *Brodatz* texture database that is publicly available online to perform and validate the proposed method. An image of 1000×1000 pixels consisting of five different Brodatz textures (Figure 6.3(a)) will be used for the unsupervised classification experimentation. Hence, we remind that our experimental data consist of the *Brodatz*, *ROI-1*, *ROI-2* data used for unsupervised classification and the *ROI-3* data used for supervised classification. In fact, our purpose of choosing these data is to demonstrate the good behavior of the proposed approach for not only the widely used natural texture database (i.e. *Brodatz*) but also the VHR satellite images (i.e. Pléiades) with both simple scenes (where textures appear quite homogeneous with clear boundaries, i.e. *ROI-1*) and complex scenes (which involve a wide variety of textures whose appearance is less homogeneous, i.e. *ROI-2* and *ROI-3*).

6.4.2 Experiment setup

The experimental procedure is activated following the description in Section 6.3 as well as the processing framework in Figure 6.2. For the evaluation of the algorithm’s performance, some reference methods are also implemented for a comparative study. We propose to exploit the local neighborhood (LN), the gray-level cooccurrence matrix (GLCM) technique [6], the Gabor filter-based descriptor (GFB) [12], the Weber local descriptor (WLD) [92], the pointwise (PW) descriptor developed previously in Chapter 4 with and without graph construction and the dense COG method [87]. The implementations of these approaches are carried out as follows:

Local neighborhood (LN) Exploit directly all pixel intensities inside the $W \times W$ dense neighborhood around each studied keypoint. Since there are total W^2 pixels, the principal component analysis (PCA) [61] is used to reduce the dimension of LN description vector. The resulted descriptor is called PCA-LN and its final dimension is equal to the number of principal components we would like to retain. In this work, it is fixed to 16 in order to ensure the compromise between the memory load and the precision of the approach. Experiments showed that about 96% to 99% of information is captured by 16 principal components.

Cooccurrence matrix (GLCM) [6] From the $W \times W$ neighborhood around each keypoint, compute four cooccurrence matrices along four main directions (0° , 45° , 90° and 135°) with the distance between pairwise pixels set to 1 and the number of gray levels set to 8, then extract five Haralick textural parameters from each matrix including the *contrast*, *correlation*, *homogeneity*, *energy* and *entropy* in order to create the 20-feature GLCM descriptor.

Gabor filter descriptor (GFB) [12] Perform the Gabor filter bank on the image by setting the number of scales to 3 and the number of orientations to 8. The window size of 2-D filter

kernel is equal to $W \times W$ pixels. Then, 24 features from the filter responses are adopted to create GFB descriptor for each keypoint.

Weber local descriptor (WLD) [92] Following the related paper, the differential excitation ξ and the quantized gradient orientation Φ for the image are first calculated using the 3×3 neighborhood. A 2-D histogram $\mathcal{H}(\xi, \Phi)$ is constructed for the $W \times W$ window around each keypoint. Then, the 1-D WLD descriptor is generated by setting $M = 6$, $T = 4$ and $S = 3$ (dedicated parameters of WLD, see [92]). Therefore, the dimension of WLD is 72.

Pointwise descriptor (PW) Generate the pointwise descriptor (proposed in Chapter 4) consisting of 12 features including the mean and variance of intensities, the mean and variance of distances, the concentration and dispersion of orientations derived from the neighboring local maximum set $\mathcal{N}_W^{\max}(p)$ and neighboring local minimum set $\mathcal{N}_W^{\min}(p)$ around each keypoint p .

Pointwise graph-based descriptors (PW+SGWT and PW+SGC) Construct a graph structure to connect all keypoints based on the similarity between PW descriptors. Then, two techniques including the spectral graph wavelet transform (SGWT) and the spectral graph clustering (SGC) are exploited to generate textural features. As mentioned in Chapter 4, the feature dimension of PW+SGWT is equal to the number of graph wavelet scales while that of PW+SGC is equal to the number of classes.

Dense COG descriptor (DCOG) [87] Construct the COG matrix for each keypoint by using all feature vectors of pixels inside the $W \times W$ dense neighborhood around it, according to Equation (6.2). Since only one 6×6 symmetrical COG matrix is constructed, the dimension of DCOG feature vector is 21 (i.e. only the entries from the upper or lower triangular part of the matrix are stored). We remind that the proposed PW-COG descriptor length is 42 since it consists of two COG matrices, i.e. one generated from $\mathcal{N}_W^{\max}(p)$ and one from $\mathcal{N}_W^{\min}(p)$.

Table 6.1 – Parameter setting in our implementation for unsupervised classification.

Parameter	Brodatz 1000 × 1000 pixels	ROI-1 900 × 1100 pixels	ROI-2 800 × 1300 pixels
ω_1	5 × 5	5 × 5	5 × 5
$\Rightarrow \begin{cases} S_{\omega_1}^{\max}(I) \\ S_{\omega_1}^{\min}(I) \end{cases}$	24469 22737	22408 20636	20150 18596
ω_2	11 × 11	11 × 11	11 × 11
$\Rightarrow S = S_{\omega_2}^{\max}(I) $	6844	6621	7106
W	50 × 50	50 × 50	50 × 50

Table 6.1 provides the parameter values that are set for the proposed algorithm during our implementation. For all data sets, we use a 5×5 search window ($\omega_1 = 5$) for the extraction of the local maximum set $S_{\omega_1}^{\max}$ and the local minimum set $S_{\omega_1}^{\min}$. A window of 11×11 pixels ($\omega_2 = 11$) is used to extract the max keypoint set S . Here, we set $\omega_2 > \omega_1$ to accelerate the computational time and $\omega_2 = 11$ provides a sufficient density and good distribution of keypoints on each image. One may probably set a smaller (resp. greater) ω_2 for a denser (resp. coarser) representation. It should be noted that when ω_2 increases, there is no influence on the texture description and

classification performance of PW-COG descriptor, only a subsampled result is obtained. On the other hand, the parameter ω_1 has a significant impact on the descriptor performance since the local max and local min pixels are included into the computation process, (see Equations (6.5) and (6.6)). Next, for the generation of the non-dense COG matrices, local maximum and minimum pixels around each keypoint are searched inside a neighborhood of size 50×50 pixels ($W = 50$). To perform an equivalent comparison, the same window size of 50×50 pixels will be set for the reference approaches including the PCA-LN, GLCM, GFB, WLD, PW and DCOG methods. For the two graph-based methods, dedicated parameters are chosen as in Chapter 4 with the number of graph wavelet scales fixed to 4 and the number of strongest connections to each graph vertex set to 1% of the number of vertices. We note that the sensitivity of the proposed method to parameters ω_1 , ω_2 and W will be studied in details later in Section 6.4.4.

After obtaining all texture descriptors from different methods for each keypoint, the final classification stage is carried out. For unsupervised classification, the K-means clustering [47] is effectuated. It is worth noting that other algorithms can be also employed such as the ISODATA, hierarchical clustering, the Fuzzy C-means, etc. [93]. Here, the K-means strategy is adopted thanks to its simplicity and good performance during our experimentation. Hence, only the number of classes is required for this stage. It is set to 5 for *Brodatz* and *ROI-1*, and 6 for the *ROI-2*. These numbers of classes in *ROI-1* and *ROI-2* are fixed for the best land-cover interpretation of each scene, in our point of view. In terms of distance metric used to perform the K-means algorithm, the classical Euclidean distance measure is exploited for the seven reference methods including PCA-LN, GLCM, GFB, WLD, PW, PW+SGWT and PW+SGC descriptors. For the two COG-based approaches (i.e. DCOG and PW-COG), we propose to investigate both the Euclidean metric and the Riemannian metric. Our objective is to demonstrate a better behavior from the latter which is based on the positive definite structure of COG matrices, as previously discussed in Section 6.2.3.

Next, for the supervised classification experiment on *ROI-3* data, some patches are extracted for the training phase. Six thematic classes are considered from the scene including the urban zones (UB01), pine forests (PF02), dense and sparse plantations (DP03 and SP04), grass fields (GF05) and bare soils (BS06). The training patches and color code associated to each class are displayed in Figure 6.6(b). We note that the size of each training patch is 150×150 pixels. For classification phase, the Minimum Distance (MD) classifier is performed using the Mahalanobis distance for GLCM, GFB and WLD descriptors, and the Riemannian distance for the DCOG and our PW-COG descriptor. Other supervised classifiers such as the K-nearest neighbors, the Support Vector Machine (SVM), etc. [94] can be also examined for this task but the simple MD classifier is selected since it has provided efficient performance during the experimentation on the studied data set. We evaluate the classification performance with the help of some validation ROIs dedicated to each class (i.e. marked in corresponding color), as shown in Figure 6.6(c). Note that training patches are excluded from validation masks. Moreover, to enrich the experimental study on this supervised classification task, the wavelet-based techniques recently proposed in [95, 96] are also investigated. More details about these approaches can be consulted in the related papers. Here, the three different models of wavelet coefficients including the Multivariate Gaussian model (MGM), the Spherically Invariant Random Vectors (SIRV) and the Gaussian Copula-based model (GCM) are performed on our *ROI-3* data for comparison.

Last but not least, quantitative results compared with ground truth may be necessary for a better assessment of each descriptor performance. The ground truth of *Brodatz* texture database is available and can be seen in Figure 6.3(b). On the contrary, ground truths of the processed Pléiades image data are not available at the moment. Hence, they are manually generated in order to bring the best land-cover texture interpretation for each studied ROI in our point of

view. The manual classified ground truths are displayed next to their input images in Figures 6.4(a)-(b) for *ROI-1* and in Figures 6.5(a)-(b) for *ROI-2*. Then, to perform the quantitative comparison, we propose to compute the overall classification accuracy (OCA) and the *Kappa* coefficient (κ) of each method. As previously defined in Section 4.4 (Chapter 4), OCA is the percentage of keypoints which are correctly classified according to the labeled keypoints from the ground truth.

6.4.3 Results and discussion

In this section, we provide unsupervised classification results for *Brodatz*, *ROI-1* and *ROI-2* data as well as supervised classification results for *ROI-3* data yielded by the proposed algorithm compared to reference methods. Qualitative and quantitative assessment is conducted in terms of classification accuracy and computational complexity. Then, the sensitivity of the proposed method to its parameters and to the distance measure will be discussed in details.

6.4.3.1 Performance in scene interpretation by texture characterization and discrimination

a) Results on unsupervised classification Figures 6.3, 6.4 and 6.5 show the unsupervised classification results of different methods applied to the *Brodatz*, *ROI-1* and *ROI-2* data, respectively. The numbers of keypoints extracted to perform these non-dense image representation and classification results are respectively 6844, 6621 and 7106 (see Table 6.1). We observe that these numbers ensure a good density and distribution of keypoints for qualitative evaluation of each result. For quantitative comparison, Table 6.2 provides the overall classification accuracy (OCA) and the *Kappa* coefficient of each method. According to the table, the best classification performance for all studied data comes from the proposed PW-COG descriptor. OCA attains the value of 96.65%, 94.94% and 91.05% while *Kappa* reaches 0.9580, 0.9234 and 0.8854 for *Brodatz*, *ROI-1* and *ROI-2*, respectively. In general, most of the methods achieve higher OCA (and *Kappa*) for the *Brodatz* and *ROI-1* than for *ROI-2* since the two former data involve more homogeneous texture zones compared to the latter. Despite this, the proposed algorithm is able to provide very effective and competitive results for all of them. Let us now investigate and analyze the classification performance of all descriptors on each data set in details, noting that the analysis will be mainly focused on the main VHR Pléiades data.

Table 6.2 – Comparison of descriptor computation time, overall classification accuracy (OCA) and *Kappa* coefficient of different methods to perform unsupervised classification on *Brodatz*, *ROI-1* and *ROI-2* data.

Descriptor	Vector length	Brodatz			ROI-1			ROI-2		
		Time (s)	OCA (%)	<i>Kappa</i>	Time (s)	OCA (%)	<i>Kappa</i>	Time (s)	OCA (%)	<i>Kappa</i>
PCA-LN	16	9.58	45.60	0.3110	8.94	79.35	0.7108	9.92	66.00	0.5760
GLCM	20	14.12	84.94	0.8118	13.78	65.97	0.5554	14.93	60.01	0.4922
GFB	24	2.46	81.24	0.7649	2.25	70.32	0.5779	2.37	73.21	0.6559
WLD	72	9.60	87.61	0.8455	9.16	52.45	0.3883	9.64	45.76	0.3420
PW	12	21.38	89.20	0.8649	19.44	80.14	0.7245	19.63	42.56	0.2899
PW+SGWT	4	70.34	86.59	0.8325	64.12	91.13	0.8620	68.73	55.19	0.4388
PW+SGC	*5(6)	76.51	94.90	0.9360	68.62	91.60	0.8693	71.10	45.86	0.3334
DCOG	21	159.16	93.07	0.9128	152.15	93.84	0.9074	164.02	77.81	0.7236
PW-COG	42	5.18	96.65	0.9580	4.74	94.94	0.9234	4.65	91.05	0.8854

(*) The vector length of PW+SGC descriptor equals to the number of classes which is set to 5 for *Brodatz* and *ROI-1*, and 6 for *ROI-2*.

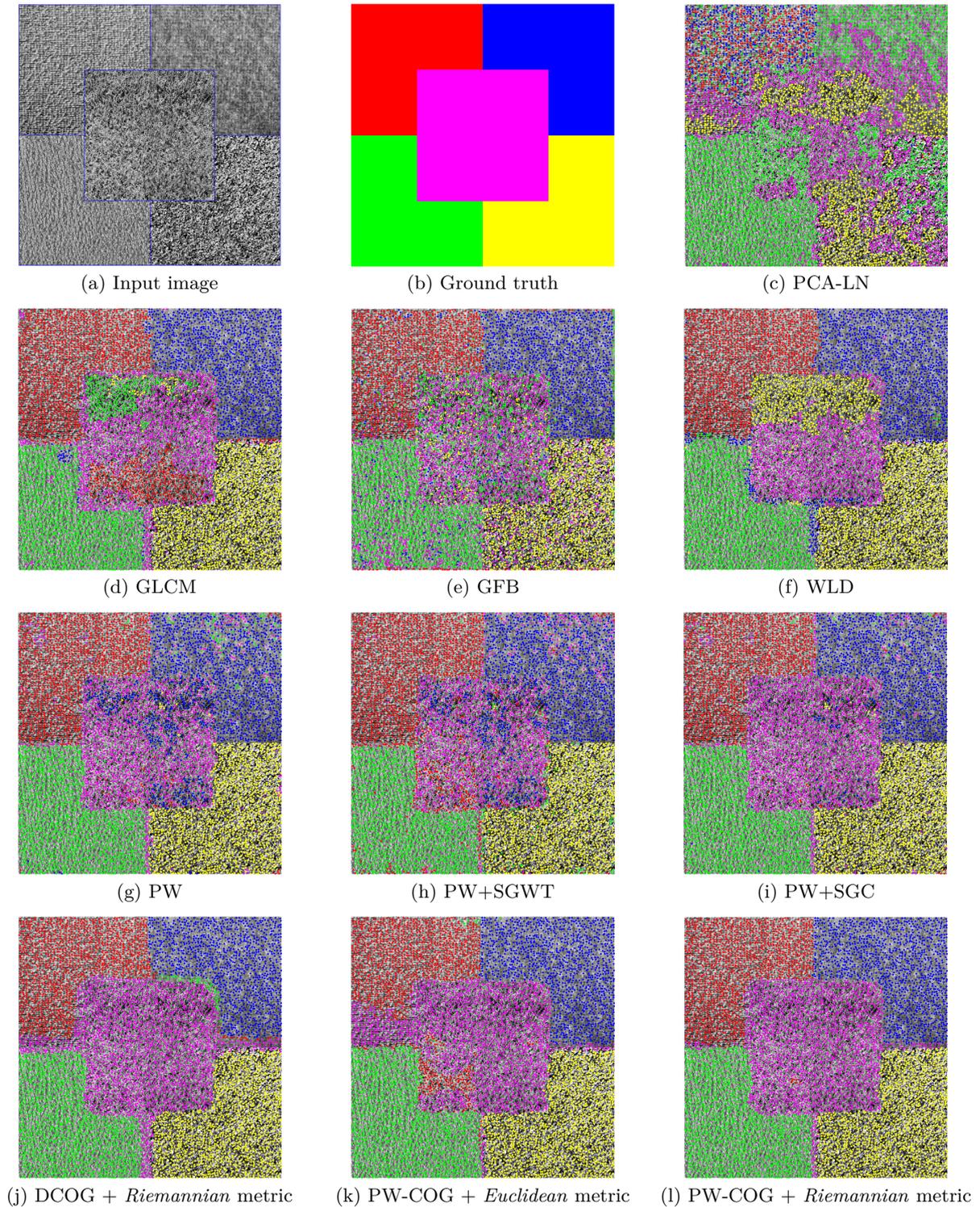


Figure 6.3 – Unsupervised classification results for *Brodatz* data. (a) Input image (1000×1000 pixels). (b) Ground truth. (c)-(l) Results yielded by the PCA-LN, GLCM, GFB, WLD, DCOG, PW+SGWT, PW+SGC, DCOG and the proposed PW-COG descriptors. The DCOG and PW-COG approaches in (j) and (l) exploit the *Riemannian* metric while the other methods employ the *Euclidean* metric for K-means clustering. (l) is our proposed method.

Brodatz data: Regarding to the ground truth from Figure 6.3(b) where the five Brodatz textures are marked in distinct colors, the proposed method (Figure 6.3(l)) yields the best classification performance with 96.65% of OCA and $\kappa = 0.9580$. We observe a good result with very few misclassified keypoints and very fine boundaries between textured regions. Other methods including the GFB, GLCM and WLD (Figures 6.3(d)-(f)) provide an OCA between 80% and 90% but involve a lot of misclassified points. PW approaches with and without graph construction (i.e. PW, PW+SGWT and PW+SGC) in Figures 6.3(g)-(i) also have good reaction to this data set. However, their results still contain many noisy points. DCOG reaches a high OCA of 93.07% ($\kappa = 0.9128$), but it still produces more errors than PW-COG. In particular, although the same window size $W \times W$ is set to generate all descriptors, this dense approach results very large borders between regions compared to our pointwise strategy, which emphasizes the efficiency of the proposed method. Moreover, we will analyze later the significant benefit of PW-COG descriptor in terms of complexity compared to the classical dense approach.

ROI-1 data: There are five expected texture classes present in this scene. From the manual ground truth in Figure 6.4(b), we mark the pine forest class in red, the two oriented vineyard classes in green and blue, the bare soil class in magenta and the road class in black. Unsupervised classification results yielded by the proposed algorithm and some reference methods are shown from Figures 6.4(c) to 6.4(l). Visually, we observe very good performance given by the DCOG and the proposed PW-COG descriptors for this scene (Figures 6.4(k)-(l)). Good thematic coherence and homogeneity are observed here in terms of texture representation and discrimination with regard to the selected keypoints. The other methods including PCA-LN, GLCM, GFB and PW produce limited results with a lot of misclassified keypoints (Figures 6.4(c)-(f)). Graph-based methods (Figures 6.3(g)-(h)) yield smooth results but they can not distinguish the blue and magenta classes. Among the five textures, only the pine forest class (red) can be characterized by all descriptors thanks to its low level of intensity. The four remaining classes involve a similar intensity level so that geometric information becomes essential to discriminate them. From Table 6.2, the proposed PW-COG descriptor provides slightly better OCA (and also *Kappa* coefficient) than the classical dense approach DCOG (94.94% compared to 93.84%) for this ROI, but we will study later its much better performance for *ROI-2* and especially its considerable benefit in terms of computational time. Four reference methods including PCA-LN, GLCM, GFB and WLD provide an OCA varying from 52% to 80%, which does not satisfy the classification purpose for this scene. The failure of GLCM approach when applied to VHR satellite data has been previously anticipated since it requires the stationarity hypothesis, at least locally stationary inside the $W \times W$ sliding window, for estimating Haralick parameters which are indeed derived from the first and second order statistical features. Next, PCA-LN approach only uses pixel intensities without any spatial and structural information so that it can not provide sufficient performance on texture characterization, even though the neighborhood window is extended to 50×50 pixels. GFB and WLD descriptors also can not provide good results applied to this studied data. Next, graph-based approaches PW+SGWT and PW+SGC significantly improve the performance of PW descriptor with an increase of about 10% in OCA. As discussed in Chapter 4, graph-based model encodes the keypoint inter-connections to enhance the PW capacity. However, compared to the proposed approach, they are still less efficient to tackle oriented texture areas within this context.

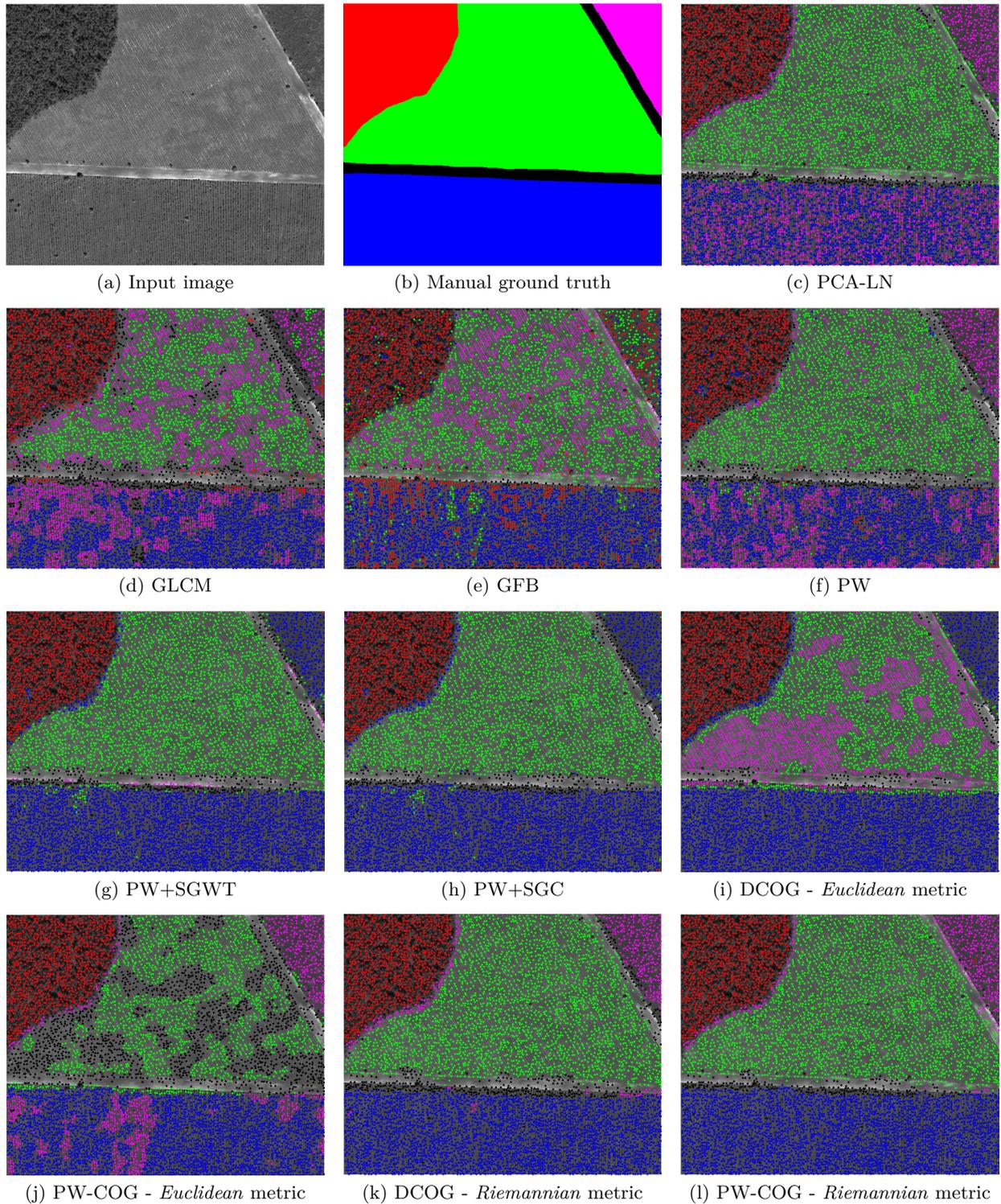


Figure 6.4 – Unsupervised classification results for *ROI-1*. (a) Input ROI (900×1100 pixels). (b) Manual ground truth. (c)-(h) Results yielded by the PCA-LN, GLCM, GFB, PW, PW+SGWT and PW+SGC descriptors. (i)-(j) Results yielded by DCOG and PW-COG methods using the *Euclidean* metric for K-means clustering. (k)-(l) Results yielded by DCOG and PW-COG methods using the *Riemannian* metric for K-means clustering. (l) is our proposed method.

ROI-2 data: Compared with the two previous ones, this ROI involves a more complex scene where textures do not appear quite separately. For example, the pine forest class that we mark in green in the ground truth in Figure 6.5(b) is present within four separate zones, or the blue grass-pasture class does not appear continuously but is corrupted by a green pine zone in the middle. Other challenges that may be listed here could be the width of the road class (marked in cyan) which is too small compared to the sliding window size used to generate all descriptors, or the unclear intersection between the yellow vineyard class and green pine zone, etc. Despite the aforementioned issues, the proposed PW-COG approach again provides very good classification result in Figure 6.5(l). GLCM, GFB as well as the graph-based PW techniques almost fail to this scene. Their results in Figures 6.5(c)-6.5(h) contain many mixtures of classes. They are not capable of discriminating separated textural classes and oriented texture zones from the scene. Next, the DCOG approach (Figure 6.5(k)) can separate the classes but it takes into more consideration the boundaries between classes and especially can not localize the bare soil class (blue). Therefore, it produces an over-classified blue class which reduces its classification accuracy. Another visual remark is that the width of the cyan road provided by the PW-COG algorithm in Figure 6.5(l) is thinner than those yielded by the GLCM method in Figure 6.5(d) and the DCOG descriptor in Figure 6.5(k), although the same window size of 50×50 pixels is used for all the three methods. Also, PW-COG delivers very fine boundaries between textured zones (as previously remarked for *Brodatz* data). That means the proposed approach is more robust to deal with local textures whose size remains too small. In terms of OCA (and *Kappa*) performance, similar behavior to the first ROI is achieved. From Table 6.2, most reference methods provide very poor results with low OCA (i.e. lower than 70%). The explication for their failure is similar as in the first ROI. OCA and *Kappa* yielded by the proposed PW-COG approach are considerably improved. They reach 91.05% and 0.8854, respectively. According to the ground truth, our pointwise method is much better than the dense strategy whose OCA and *Kappa* are 77.81% and 0.7236, which again confirms the superior performance of the proposed strategy for this kind of VHR satellite images.

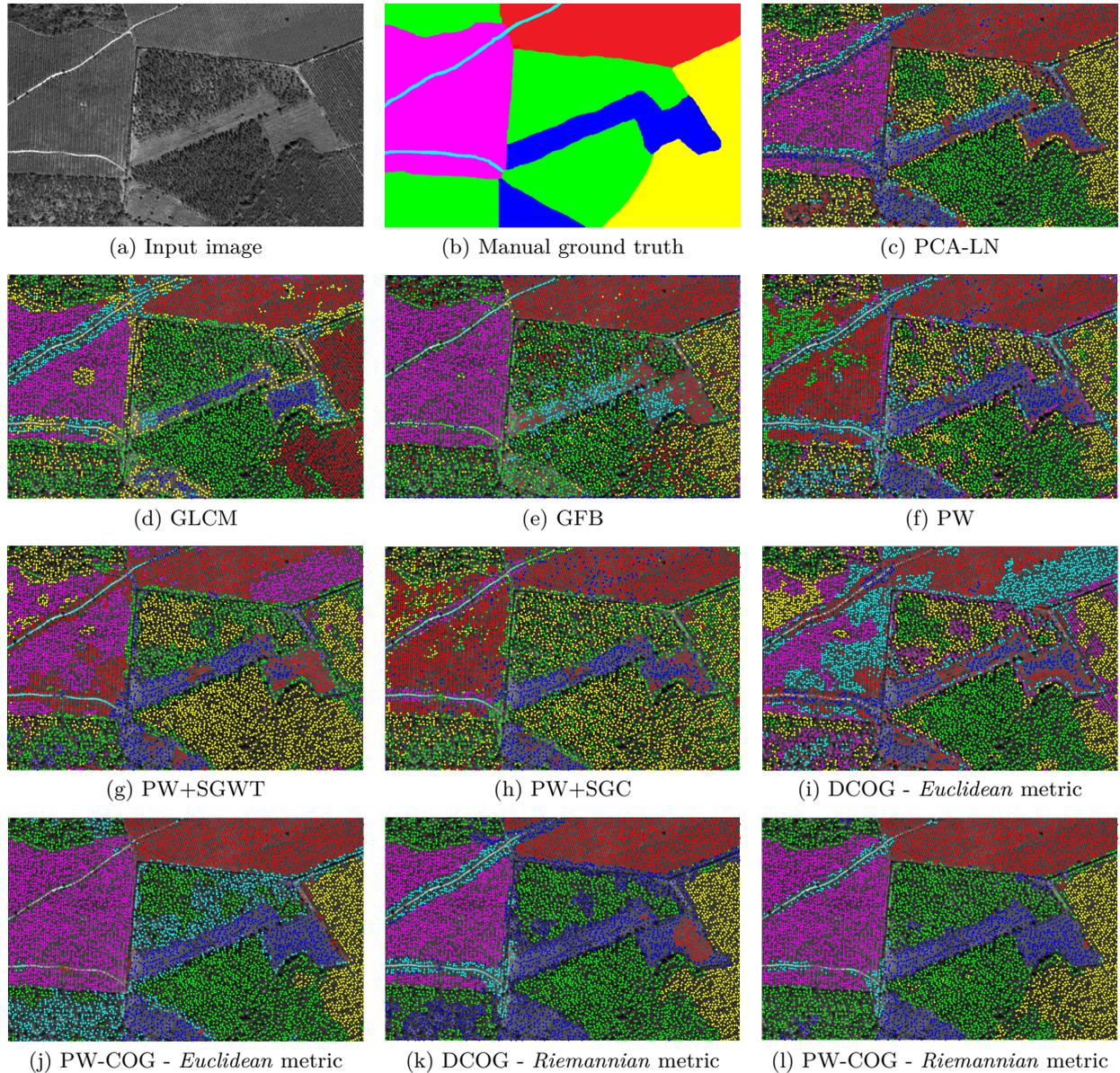


Figure 6.5 – Unsupervised classification results for *ROI-2*. (a) Input ROI (800×1300 pixels). (b) Manual ground truth. (c)-(h) Results yielded by the PCA-LN, GLCM, GFB, PW, PW+SGWT and PW+SGC descriptors. (i)-(j) Results yielded by DCOG and PW-COG methods using the *Euclidean* metric for K-means clustering. (k)-(l) Results yielded by DCOG and PW-COG methods using the *Riemannian* metric for K-means clustering. (l) is our proposed method.

b) Results on supervised classification Table 6.3 reports the supervised classification performance of different methods performed on *ROI-3* data according to the validation masks shown in Figure 6.6(c). Here, the computational time for both training and classifying phase is delivered. In terms of classification accuracy, we provide the performance of each method for each class from the scene, the average and overall classification accuracy (i.e. ACA and OCA) and the *Kappa* coefficient. OCA has been previously defined in Section 6.4.2 while ACA is computed by averaging all the classification accuracy for each class. We note that the results of three wavelet-based models (i.e. MGM, SIRV, GCM) have been provided by the authors of [95, 96] by processing the *ROI-3* data using the same training and validation zones as well as equivalent parameter setting to our approach. Also, although these methods are developed using a dense approach, their classification performance is re-computed on keypoint positions only for a fair comparison. Since the keypoint’s definition in our work (i.e. using local maxima) has a good capacity of image representation, the results of these three methods on keypoints do not change significantly from their original dense approach.

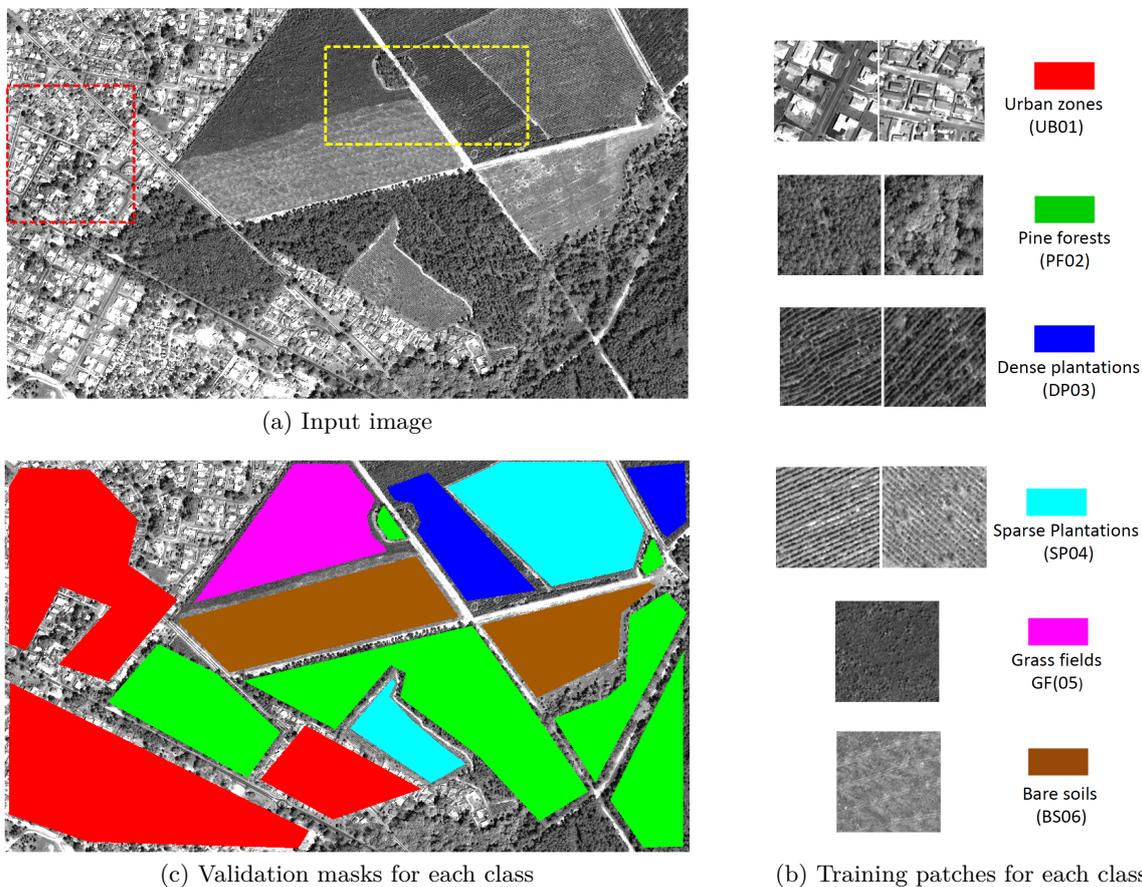


Figure 6.6 – Supervised classification experiment for *ROI-3*. (a) Input ROI (2000×3200 pixels) in which the red and yellow dashed rectangles involve two interesting cuts (i.e. *Red cut* and *Yellow cut*) that will be analyzed in Figure 6.7. (b) Training patches and color label used for each class; each patch size is 150×150 pixels. (c) Validation masks (with corresponding color) for each class.

Similar to the previous unsupervised classification experimentation, the best performance is again adopted by the proposed PW-COG method whose ACA, OCA and *Kappa* reach 92.05%, 90.01% and 0.8767, respectively. From the table, GFB and WLD seem to be not relevant for this scene with the accuracy lower than 65%. Next, the behavior of DCOG here is again similar

Table 6.3 – Comparison of computation time and classification accuracy of different methods to perform supervised classification on *ROI-3* data.

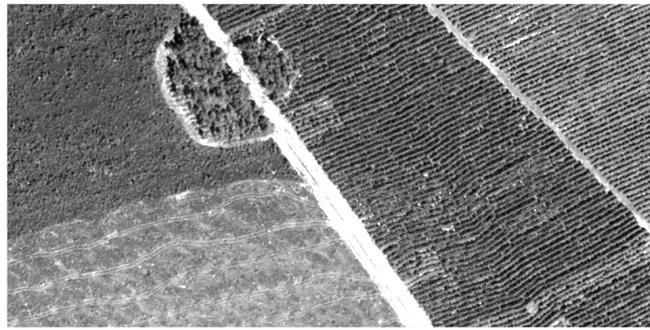
Descriptor	Time (s)		Accuracy for each class (%)						Performance		
	Training	Classifying	UB01	PF02	DP03	SP04	GF05	BS06	ACA (%)	OCA (%)	<i>Kappa</i>
GFB	12.82	38.98	56.91	53.73	35.00	82.12	44.88	78.16	58.47	60.46	0.5128
WLD	41.62	197.03	98.89	72.39	66.19	75.03	14.68	6.91	55.68	63.05	0.5259
GLCM	4.54	113.34	99.72	86.13	84.34	85.72	87.36	56.92	83.37	84.81	0.8101
MGM	5.00	243.00	86.63	89.30	88.98	96.15	69.17	72.90	83.86	85.41	0.8113
SIRV	39.00	289.00	87.28	96.01	84.07	97.27	61.21	71.63	82.91	85.97	0.8232
GCM	10.00	358.00	87.98	88.90	81.37	89.98	67.42	77.00	82.11	84.53	0.8059
DCOG	92.78	260.55	83.22	82.42	87.35	97.62	97.27	89.63	89.58	88.05	0.8529
PW-COG	4.37	26.12	82.51	84.73	93.14	96.86	98.52	96.53	92.05	90.01	0.8767

to unsupervised classification task with its good but lower performance (i.e. OCA= 88.05% and $\kappa = 0.8529$) than PW-COG and its high computational cost. GLCM and three wavelet-based techniques achieve quite good performance with an OCA varies between 84% and 85%. Especially in terms of class performance, GLCM has the best accuracy for the urban class (UB01) while the SIRV model provides the best one for the pine forest class (PF02). The reason why GLCM yields the best performance on UB01 and quite good result on PF02 is that the validation masks selected for these class (red mask for UB01 and green mask for PF02, see Figure 6.6(c)) are considered to be homogeneous (for thematic interpretation). However, within these urban regions, there exist many local textures (i.e. several trees inside urban zones, small road from forest areas, etc.) which potentially cause heterogeneous classification results. Our experiments show that the PW-COG takes into more account these local textures. Hence, within its classification result, a lot of local features are detected (green labels inside urban zone UB01, brown or magenta labels inside forest zone PF02, etc.). We demonstrate this remark in Figures 6.7(a)-(b)-(c) where classification results of an urban area (i.e. *Red* cut from *ROI-3*) yielded by the GLCM and PW-COG methods are displayed. We observe that from the PW-COG result (Figure 6.7(c)), some trees are detected and marked in green, which reduces the accuracy for UB01 class. On the other hand, the GLCM provides very uniform (i.e. homogeneous) result in Figure 6.7(b) which produces the highest performance for UB01 in Table 6.3. From this point, despite its lower accuracy for UB01 and PF02 classes, the proposed approach is still considered to be very effective for local textures and becomes a better candidate for applications which take into account local features within VHR images.

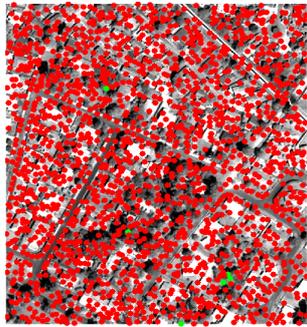
For other classes involving more homogeneous textures such as plantations (DP03, SP04), grass fields (GF05) or bare soils (BS06), the PW-COG method provides very efficient performance which considerably improves its OCA and *Kappa*. An illustration is shown in Figures 6.7(d)-(e)-(f) including the classification results of the GLCM and PW-COG for the *Yellow* cut from *ROI-3*. Here, much better performance can be observed from and validated for the proposed approach (Figure 6.7(f)) compared against the GLCM (Figure 6.7(e)). In conclusion, all of the results shown in Table 6.3 and Figure 6.7 have demonstrated and confirmed the effectiveness of the proposed PW-COG method applied to supervised classification on this *ROI-3* data set.



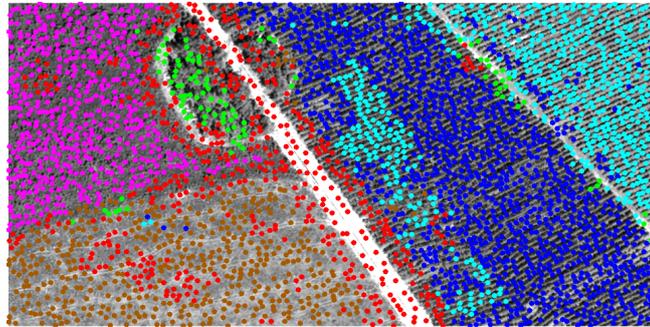
(a) *Red cut* from *ROI-3* data



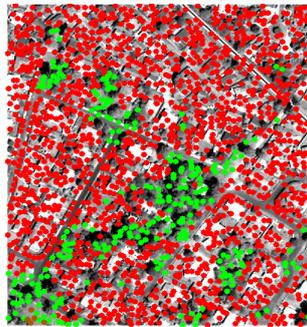
(d) *Yellow cut* from *ROI-3* data



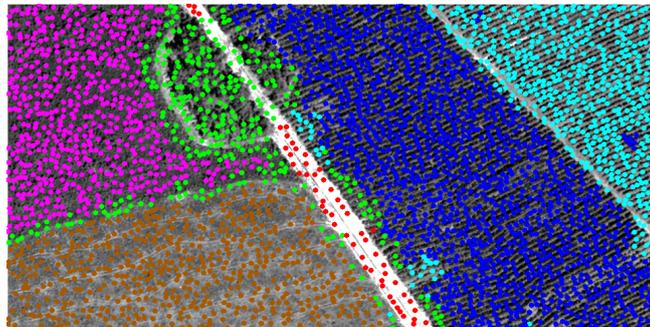
(b) *Red cut* - GLCM



(e) *Yellow cut* - GLCM



(c) *Red cut* - PW-COG



(f) *Yellow cut* - PW-COG

Figure 6.7 – Supervised classification results for the *Red* and *Yellow* cuts from *ROI-3* (see Figure 6.6 for their locations). (a)-(c) Input, classification results by GLCM and PW-COG methods for *Red* cut. (d)-(f) Input, classification results by GLCM and PW-COG methods for *Yellow* cut. Colored labels of different classes are defined as in Figure 6.6.

6.4.3.2 Performance in time consumption

In terms of algorithm complexity, the proposed pointwise COG strategy is much faster than the traditional dense COG approach. We note that all implementations are effectuated using MATLAB on a machine *Xeon 3.6GHz, 16GB RAM*. From unsupervised classification results in Table 6.2, the computational time of PW-COG algorithm (about 5s in average for each data set) is significantly reduced compared to the DCOG method (about 160s). Similarly for supervised classification result in Table 6.3, both training and classifying time of PW-COG (30.5s) is again much lower than DCOG (353.3s). As mentioned in Section 6.2, with a given window size $W \times W$ pixels used to generate COG matrices, our PW-COG approach only takes into account feature vectors located at local maxima in \mathcal{N}_W^{\max} and at local minima \mathcal{N}_W^{\min} while DCOG considers all W^2 feature vectors. Since $|\mathcal{N}_W^{\max}| + |\mathcal{N}_W^{\min}| \ll W^2$, the PW-COG calculation time is considerably lower. As anticipated, the high computational cost within the two graph-based methods (i.e. PW+SGWT and PW+SGC) is caused by graph construction stage while the three wavelet-based approach (i.e. MGM, SIRV and GCM) require a lot of time to generate their multivariate models for multiscale wavelet coefficients. Next, for the other methods, only the GFB (about 2.3s in average) is faster than our algorithm in case of unsupervised classification (Table 6.2) while all the remaining ones (i.e. PCA-LN, GLCM, WLD, PW) require more calculation time. Then, for the supervised classification task, since the complexity of the Mahalanobis distance within the Minimum Distance classifier is more costly than that of the Riemannian distance (only applied to class centers), the proposed approach becomes the lowest-complexity one. Therefore, the efficiency of our strategy is also confirmed in terms of time consumption.

Another remark is that the complexity of dense approaches (PCA-LN, GLCM, WLD, GFB and DCOG) algorithm proportionally increases with the number of representative keypoints. Hence, a higher computational time is required in case that more keypoints are extracted for the image representation and classification. On the other hand, the complexity of PW-COG algorithm will depend on both the number of keypoints and the number of local max and min pixels (characterized by parameters ω_2 and ω_1). We observe this dependence on these parameters for details in the next subsection.

6.4.4 Sensitivity to parameters

The proposed PW-COG method involves three parameters: the window size used for the extraction of local maximum and local minimum pixels (ω_1), the window size for the extraction of representative max keypoints (ω_2) and the window size set for the generation of COG matrix descriptor (W). The sensitivity of the approach to these parameters is experimented on *ROI-1* data and shown in Tables 6.4, 6.5 and 6.6. We would like to emphasize that the algorithm's performance depends considerably on ω_1 and W , but insignificantly on ω_2 . In fact, during the generation of $\delta^{\text{PW-COG}}$ in Section 6.2.2, ω_2 is not involved (see Equations (6.5) and (6.6)). A smaller or higher ω_2 just provides a denser or coarser keypoint-based image representation and classification. As observed from Table 6.5, classification performance (i.e. OCA and *Kappa*) is quite stable when ω_2 varies. Only the number of keypoints (and hence the computation time) will be affected. It may be chosen from 7×7 to 11×11 to ensure a good density of keypoints as well as a low calculation time.

On the contrary, W is directly employed to search for the closest local max and closest local min pixels around each keypoint during construction of COG matrices, and ω_1 is linked to the density of these local extrema. Indeed, ω_1 represents the scaling factor of the pointwise approach using local max and min pixels as characteristic points to approximate and characterize textures. During the experiments on VHR Pléiades images with 50cm resolution, we observe

Table 6.4 – Sensitivity of the proposed method to the window size ω_1 . Experiments are performed on *ROI-1* data.

Window ($\omega_1 \times \omega_1$)	Number of maxima	Number of minima	Time (s)	OCA (%)	<i>Kappa</i>
3×3	35944	34928	6.61	94.74	0.9204
5×5	22408	20636	4.68	94.94	0.9234
7×7	13991	13934	3.61	95.03	0.9246
9×9	9426	11042	3.05	92.45	0.8842

Table 6.5 – Sensitivity of the proposed method to the window size ω_2 . Experiments are performed on *ROI-1* data.

Window ($\omega_2 \times \omega_2$)	Number of keypoints	Time (s)	OCA (%)	<i>Kappa</i>
Dense	900×1100	682.98	93.15	0.8995
3×3	35944	25.22	94.73	0.9189
7×7	13991	9.98	94.84	0.9215
11×11	6621	4.71	94.94	0.9234
15×15	3424	2.45	95.15	0.9275

Table 6.6 – Sensitivity of the proposed method to the window size W . Experiments are performed on *ROI-1* data.

Window ($W \times W$)	Time (s)	OCA (%)	<i>Kappa</i>
35×35	2.39	92.13	0.8787
40×40	3.16	91.95	0.8765
45×45	3.74	95.26	0.9279
50×50	4.74	94.94	0.9234
55×55	5.42	94.64	0.9190
60×60	6.52	91.29	0.8686

that a window ω_1 set from 3×3 to 7×7 can yield good performance for the proposed strategy (see Table 6.4). Increasing ω_1 too much may decrease the distinctiveness capacity of the descriptor. The remaining parameter W does exist in most of local descriptors. It represents the level of exploiting information from neighboring pixels to support the construction of descriptor at each given pixel. Table 6.6 shows the sensitivity of the method to W . We observe that for our data, good performance of the proposed descriptor can be achieved when W varies from 35 to 60, particularly the best should be adopted from 45 to 55.

6.4.5 Sensitivity to the distance metric

Table 6.7 provides the classification results given by the two COG-based descriptors (i.e. DCOG and PW-COG) using two different distance metrics within the K-means clustering stage, performed on *ROI-1* and *ROI-2* data sets. We propose this comparison to confirm the benefit of exploiting the geometric-based Riemannian metric when using COG-based descriptors, compared to the classical Euclidean metric. Visual results for the two ROIs can be found in Figures

Table 6.7 – Comparison of overall classification accuracy (OCA) yielded by the DCOG and PW-COG methods by using Euclidean and Riemannian distance metrics. Experiments are performed on *ROI-1* and *ROI-2* data.

Distance metric	ROI-1		ROI-2	
	DCOG	PW-COG	DCOG	PW-COG
Euclidean	75.97%	67.29%	40.68%	75.82%
Riemannian	93.84%	94.94%	77.81%	91.05%

6.4(f) to 6.4(i) and Figures 6.5(f) to 6.5(i). As observed, the results given by the Riemannian metric are always better than those produced by the Euclidean metric, for both DCOG and PW-COG methods performed on both ROIs. For example in case of *ROI-2*, the cyan road class can not be recognized when using the Euclidean metric (Figures 6.5(f)-(g)) but it is well tracked by employing the Riemannian measure (Figures 6.5(h)-(i)). This remark can be validated by the quantitative comparison in Table 6.7. As illustrated in the table, the value of OCA is considerably increased when the Riemannian metric is adopted instead of the Euclidean metric. In particular within our PW-COG strategy, OCA is enhanced from 67.29% to 94.94% for *ROI-1* and from 75.82% to 91.05% for *ROI-2*.

6.5 Conclusion

A novel local descriptor called PW-COG has been developed to tackle the texture description task in the scope of VHR imagery by performing the pointwise approach on the construction of covariance-based descriptors. During the chapter, the local max and local min pixels have continued to be exploited to perform the pointwise approach while radiometric information and structural features (given by different image gradients) are integrated to form covariance descriptors. Then, both keypoint-based unsupervised and supervised classification tasks have been performed for the evaluation of the proposed algorithm. Experimental results on *Brodatz* texture database and on VHR panchromatic Pléiades image data show the efficiency of the proposed descriptor compared to several reference methods, including the proposed pointwise graph-based descriptors in Chapter 4. For a conclusion, this novel PW-COG descriptor satisfies our first goal of a direct and fast local descriptor (without graph construction). Secondly, in terms of texture characterization and discrimination, it has been proved to be effective to deal with oriented texture zones thanks to the integration of structural information.

The next chapter continues to exploit structural gradient features to perform a pointwise texture retrieval framework. Another local descriptor will be developed and proved to be relevant for retrieving structured textures from an image’s scene. It will be applied to tackle texture-based vine detection task in the agricultural domain.

LED: An efficient local extrema descriptor for texture retrieval from VHR images applied to vineyard detection

Problems are not stop signs, they are guidelines.

Robert H. Schuller

7.1	Introduction	120
7.1.1	Motivation of the chapter	120
7.1.2	Chapter organization	120
7.2	Texture retrieval from VHR optical images using the local extrema descriptor . .	121
7.2.1	Extraction of LED	121
7.2.2	Dissimilarity measure for retrieval	124
7.2.3	Retrieval algorithm	125
7.3	Retrieval experiments	127
7.3.1	Studied sites and data	127
7.3.2	Experiment setup	128
7.3.3	Results and discussion	129
7.3.4	Sensitivity to parameters	130
7.4	Application to vineyard detection based on supervised classification	133
7.4.1	Supervised classification algorithm	133
7.4.2	Experimental results	134
7.5	Conclusion	140

The content of this chapter is mainly based on the following published papers:

- M. T. Pham, G. Mercier, O. Regniers, J. Michel, *Texture retrieval from VHR optical remote sensed images using the local extrema descriptor with application to vineyard parcel detection*, Remote Sensing, vol. 8(5), 368, 2016.
- M. T. Pham, G. Mercier, O. Regniers, L. Bombrun, J. Michel, *Texture retrieval from very high resolution remote sensing images using the local extrema-based descriptors*, Proc. IEEE Int. Geosci. Remote Sens. Symp., Beijing, China, 2016.

7.1 Introduction

7.1.1 Motivation of the chapter

In this chapter, we carry out a study of vineyard cultivation by detecting vine parcels using VHR optical remote sensing images. In particular, our motivation is to perform texture-based retrieval and supervised classification frameworks to automatically distinguish vineyard fields from other items present from the image content such as forest zones, bare soils, early grown grasses, urban areas, etc. Some research studies have been so far developed to tackle retrieval and classification tasks in the scope of remote sensing imagery applied to agricultural fields, particularly in vineyard cultivation. Classical statistical texture analysis techniques such as the gray-level cooccurrence matrix (GLCM) [6], the Gabor filter banks (GFB) [12], the Weber local descriptor (WLD) [92] or the multiscale discrete wavelet decomposition [15] have been investigated and adapted to VHR image data. In [97], GLCM features are exploited for classification of orchards and vineyards using VHR panchromatic images. In [35], they are used to retrieve different forest structure variables from IKONOS-2 images. Next, methods based on the Gabor filter coefficients are proposed in [98,99] for vine plot detection using optical images. In [39,100], multi-resolution texture analysis using wavelet techniques is investigated on VHR remote sensing data. The WLD is not frequently used in optical remote sensing field, but efforts have been done to study its behavior on VHR polarimetric Radar data for patch indexing [101]. Among these approaches, wavelet-based techniques appear to be the most commonly used up to now. Some recent studies have focused on the modeling of wavelet coefficients using multivariate distribution models. In [95,102], the authors study the multivariate Gaussian models (MGM), the spherically invariant random vectors (SIRV) and the Gaussian copula-based models (GCM) to tackle texture retrieval and classification of VHR maritime pine forest images.

Throughout the previous chapters of this thesis work, we have witnessed the limitation of the above classical approaches applied to VHR remote sensing images. Texture-based classification experiments in Chapter 4 and Chapter 6 showed their inferior performance compared to our proposed pointwise graph-based descriptors (i.e. PW+SGC and PW+SGWT) as well as pointwise covariance descriptor (i.e. PW-COG). In the present chapter, we continue to employ the pointwise texture analysis approach (cf. Chapter 2) for the application of vine detection using VHR image data. In order to do that, we propose to integrate more structural features into the pointwise (PW) descriptor (developed in Chapter 4) to improve its capacity of texture characterization and discrimination. This novel descriptor, called the local extrema descriptor (LED), is now relevant for retrieving and detecting oriented and structured textures coming from vine fields within the image. In collaboration with Lionel Bombrun (University of Bordeaux) and Olivier Regniers (I-SEA company), we first embed the proposed descriptor into a retrieval framework to validate its performance of texture description and recognition. Then, it is exploited to tackle supervised classification task from which the main motivation is to detect vine parcels. Experimental results show the effectiveness of the proposed retrieval algorithm, particularly in retrieving structural features such as man-made items in urban zones and different types of aligned vine rows. Next, in terms of vine detection performance, the proposed strategy also provides very promising and competitive results compared to reference methods.

7.1.2 Chapter organization

The remainder of this chapter is organized as follows. Section 7.2 describes the proposed texture-based retrieval algorithm using the novel local extrema descriptor (LED). Experimental results performed on texture databases extracted from VHR optical images are shown in Section 7.3 for

the evaluation and validation of the proposed strategy. Then, in Section 7.4, the application of LED feature vectors to the detection of vine parcels is performed via a supervised classification scheme. Finally, Section 7.5 concludes the chapter's contribution.

7.2 Texture retrieval from VHR optical images using the local extrema descriptor

A content-based image retrieval framework generally consists of two primary stages: the feature extraction to characterize each image from the database and the distance measure used to retrieve the group of images having the same category as each query image. An outline can be observed in Figure 7.1. In this section, we would like to tackle retrieval task based on texture analysis. To perform our scheme, a set of local extrema descriptors (LEDs) are first extracted to characterize textural features of each image. Then, geometric-based distance metric between those LED feature sets will be employed to measure the images' dissimilarity. We now address each stage in details before providing the complete texture-based retrieval algorithm.

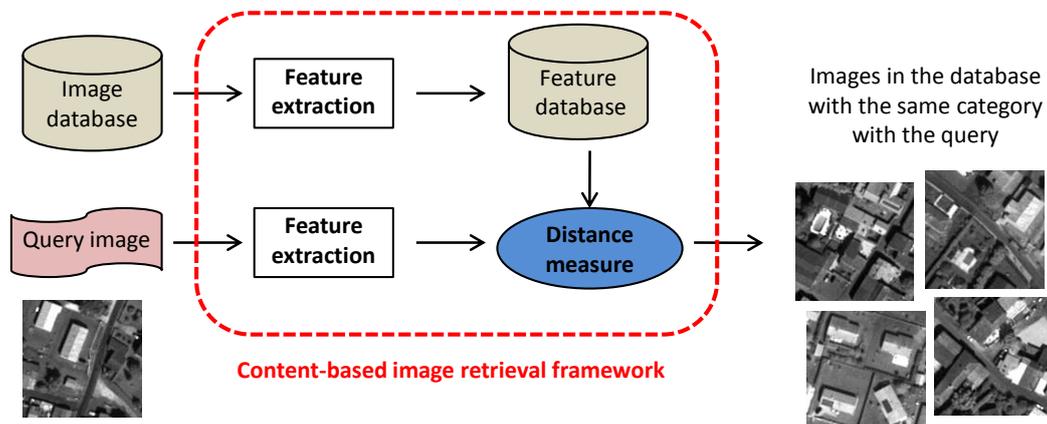


Figure 7.1 – A general content-based image retrieval framework.

7.2.1 Extraction of LED

Figure 7.2 highlights the feature extraction stage. The idea is that, for each query image, we extract a set of characteristic points (i.e. keypoints) and then generate their local descriptors. Hence, the image is encoded by a set of local descriptors which can be considered as a point cloud within the feature space. In this chapter, we continue to exploit the pointwise approach based on the local maximum and local minimum pixels which is the main core throughout the thesis work. Our proposition is to extract the local extrema descriptors (LEDs) to characterize local textures and structures around keypoints. Here, keypoints continue to be the local maximum pixels extracted using the same or a different search window size (the same principle has been applied to Chapter 4 and Chapter 6). The LED is in fact an improved version of the pointwise (PW) descriptor which was proposed in Chapter 4. It is generated by incorporating both radiometric and geometric information of the local maximum and local minimum pixels around each central keypoint. As we have to deal with oriented and structured textures given by vine plot characteristics observed from VHR images within the context of this chapter, more

structural information needs to be encoded. We propose to insert the gradient magnitude and orientation features of those local extrema into the novel LED to enhance its performance.

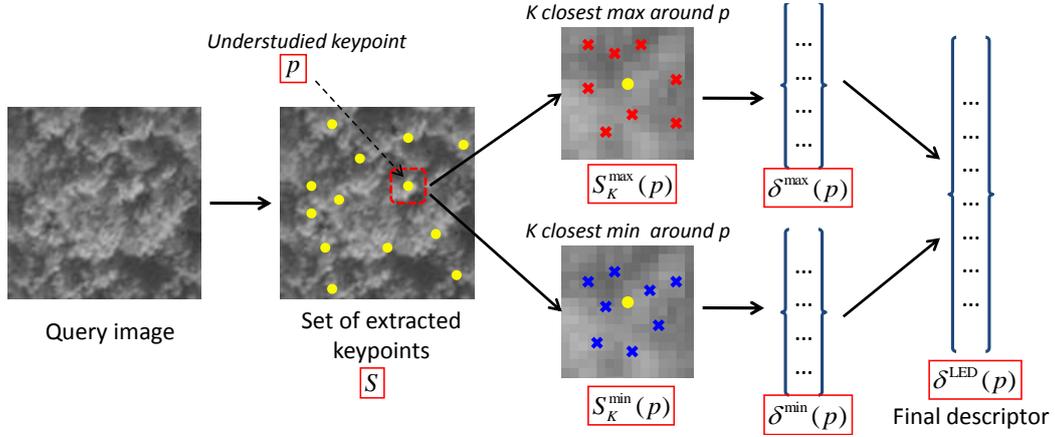


Figure 7.2 – Generation of local extrema descriptors (LEDs) for each query image.

The extraction of the local maximum and local minimum pixels from a panchromatic image can be reviewed in Chapter 2 (see Equations (2.1) and (2.2)). For a recall, the local maximum set and local minimum set extracted from the image I using the search window of size $\omega \times \omega$ pixels are denoted by $S_\omega^{\max}(I)$ and $S_\omega^{\min}(I)$, respectively. Let us consider an understudied keypoint p located at position (x_p, y_p) having its intensity $I(p)$. To extract the LED feature vector of p , we first search for a set of K closest local maxima and a set of K closest local minima around p from the two sets $S_\omega^{\max}(I)$ and $S_\omega^{\min}(I)$. These two sets are denoted by $S_K^{\max}(p)$ and $S_K^{\min}(p)$, respectively (see Figure 7.2). The following features involving radiometric (i.e. intensity), spatial and gradient information are extracted for each set. For better explanation, Figure 7.3 shows the geometric and gradient features derived from each local max or local min within these sets. Below are the features extracted from $S_K^{\max}(p)$, the feature computation for $S_K^{\min}(p)$ is similar. Readers are also invited to review the generation of PW descriptor in Section 4.2 of Chapter 4 for better understanding.

- Mean and variance of intensities computed as Equations (4.1) and (4.2):

$$\mu_I^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} I(q), \quad (7.1)$$

$$\sigma_I^2{}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (I(q) - \mu_I^{\max}(p))^2. \quad (7.2)$$

- Mean and variance of spatial distances of every point from the set to the keypoint p computed as Equations (4.3) and (4.4):

$$\mu_d^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} d(p, q), \quad (7.3)$$

$$\sigma_d^2{}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (d(p, q) - \mu_d^{\max}(p))^2, \quad (7.4)$$

where $d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$.

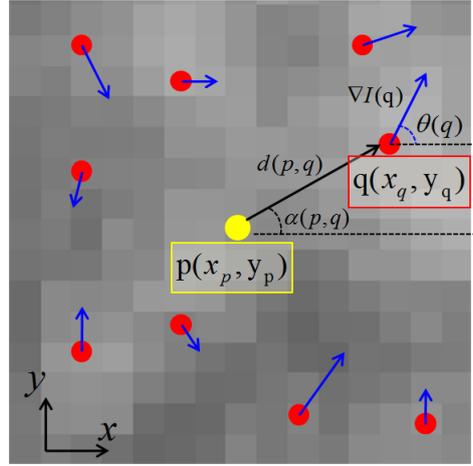


Figure 7.3 – Geometric and gradient information from a local maximum (resp. local minimum) pixel $q = (x_q, y_q)$ within $S_K^{\max}(p)$ (resp. $S_K^{\min}(p)$) considered for the calculation of LED at the studied keypoint $p = (x_p, y_p)$. Here, $d(p, q)$ is the distance between p and q ; $\alpha(p, q)$ is the angle of the vector yielded from p to q . We have $d(p, q) = d(q, p)$ but $\alpha(p, q) \neq \alpha(q, p)$. Then, $|\nabla I(q)|$, $\theta(q)$ are the gradient magnitude and gradient orientation at q .

- Measure of circular variance of angles formed by those points and p :

$$\sigma_{\text{cir}, \alpha}^2(p) = 1 - \sqrt{\overline{C}_\alpha^2(p) + \overline{S}_\alpha^2(p)}, \quad (7.5)$$

where

$$\overline{C}_\alpha(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \cos \alpha(p, q),$$

$$\overline{S}_\alpha(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \sin \alpha(p, q),$$

$$\alpha(p, q) = \arctan \left(\frac{y_q - y_p}{x_q - x_p} \right), \quad \alpha(p, q) \in [-\pi, \pi], \forall p, q.$$

- Mean and variance of gradient magnitudes:

$$\mu_{|\nabla I|}^{\max}(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} |\nabla I(q)|, \quad (7.6)$$

$$\sigma_{|\nabla I|}^2(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} (|\nabla I(q)| - \mu_{|\nabla I|}^{\max}(p))^2, \quad (7.7)$$

where $|\nabla I|$ is the gradient magnitude image obtained by applying the Sobel filtering operator on the image.

- Measure of circular variance of gradient orientations:

$$\sigma_{\text{cir}, \theta}^2(p) = 1 - \sqrt{\overline{C}_\theta^2(p) + \overline{S}_\theta^2(p)}, \quad (7.8)$$

where

$$\overline{C}_\theta(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \cos \theta(q),$$

$$\bar{S}_\theta(p) = \frac{1}{K} \sum_{q \in S_K^{\max}(p)} \sin \theta(q),$$

θ is the gradient orientation image obtained together with the gradient magnitude $|\nabla I|$ by performing the Sobel filter.

Denote $\delta^{\max}(p)$ the feature vector generated for the closest local maximum set $S_K^{\max}(p)$ around keypoint p , we have:

$$\delta^{\max}(p) = \left[\mu_I^{\max}(p), \sigma_I^2{}^{\max}(p), \mu_d^{\max}(p), \sigma_d^2{}^{\max}(p), \right. \\ \left. \sigma_{\text{cir},\alpha}^2{}^{\max}(p), \mu_{|\nabla I|}^{\max}(p), \sigma_{|\nabla I|}^2{}^{\max}(p), \sigma_{\text{cir},\theta}^2{}^{\max}(p) \right] \in \mathbb{R}^8. \quad (7.9)$$

Similar process is applied to obtain $\delta^{\min}(p)$ from $S_K^{\min}(p)$. Finally, denote $\delta^{\text{LED}}(p)$ the final LED feature vector of p , we insert the intensity $I(p)$ to form it as follows:

$$\delta^{\text{LED}}(p) = \left[I(p), \delta^{\max}(p), \delta^{\min}(p) \right] \in \mathbb{R}^{17}. \quad (7.10)$$

Being an extended version of the PW descriptor (Chapter 4), the proposed $\delta^{\text{LED}}(p)$ allows us to characterize the local environment around keypoint p by understanding how the local maxima and the local minima are distributed and arranged. By capturing the spatial information, radiometric characteristics and structural properties (given by gradient features) of these local extrema, it is able to represent and discriminate textural features, in particular to deal with oriented and structured texture zones. It is worth noting that LED is invariant to rotation. As observed from its generation, for the two directional features including the geometric angle α and the gradient orientation θ , only their directional variances [29] are taken into account, their mean values are discarded to ensure the rotation-invariant property. In terms of feature dimensionality, LED vector consists of 17 features. Like the PW descriptor, LED can be also improved or adapted by modifying or adding other features into the vector in Equations (7.9), (7.10). For example, one could insert more gradient features from different orientations, others may involve some multiscale features by certain filtering processes, etc. The improvements will depend on one's expectation to perform their own descriptor. Here, the LED is constructed basically from radiometric, geometric and gradient properties which are relevant for texture analysis within the context of vineyard detection.

7.2.2 Dissimilarity measure for retrieval

Each image patch I_i from the texture database is now characterized by a set of LEDs which can be considered as a LED feature point cloud F_i of size $N_i \times 17$, where $N_i = |S_i|$ is the number of keypoints extracted from I_i :

$$F_i = \left\{ \delta^{\text{LED}}(p) \right\}_{p \in S_i}. \quad (7.11)$$

We now compute the dissimilarity matrix involving all pair-wise distance measures for the database. Here, we propose to investigate two different distance measures: the *simplified Mahalanobis* distance and the *Riemannian* distance [91]. For its computation, the Mahalanobis distance takes into account the mean feature vector and the feature covariance matrix of each feature point cloud. On the other hand, the Riemannian distance focuses on the geometric structures of the two point clouds. Hence, it only considers the covariance matrix, not the mean

feature vector, of each point cloud during its calculation. The estimations of the mean feature vector μ_i and the feature covariance matrix C_i of a point cloud F_i are as follows:

$$\mu_i = \frac{1}{N_i} \sum_{p \in S_i} \delta^{\text{LED}}(p), \quad (7.12)$$

$$C_i = \frac{1}{N_i} \sum_{p \in S_i} \left(\delta^{\text{LED}}(p) - \mu_i \right)^T \left(\delta^{\text{LED}}(p) - \mu_i \right). \quad (7.13)$$

Let us denote μ_i , μ_j , C_i and C_j the mean vectors and the feature covariance matrices estimated from the two feature point clouds F_i and F_j , respectively. The simplified Mahalanobis distance between I_i and I_j is computed as:

$$\begin{aligned} d_{\text{mahalanobis}}(I_i, I_j) &= d_{\text{mahalanobis}}(F_i, F_j) \\ &= (\mu_i - \mu_j) \left(C_i^{-1} + C_j^{-1} \right) (\mu_i - \mu_j)^T, \end{aligned} \quad (7.14)$$

and the Riemannian distance is calculated by:

$$\begin{aligned} d_{\text{riemannian}}(I_i, I_j) &= d_{\text{riemannian}}(F_i, F_j) \\ &= \sqrt{\sum_{\ell=1}^d \ln^2 \lambda_\ell}, \end{aligned} \quad (7.15)$$

where λ_ℓ is the ℓ^{th} generalized eigenvalue which satisfies $\lambda_\ell C_i \chi_\ell - C_j \chi_\ell = 0$; $\ell = 1, \dots, d$. χ_ℓ is the corresponding eigenvector to λ_ℓ and $d = 17$ is the dimension of LED feature vector.

Once the distance matrix is formed, a cross-validation approach with random selection procedure is employed to evaluate the retrieval performance. At each iteration, an equal number of n images is randomly extracted for each class. Based on the distance matrix, n most similar images are considered for each query image. The retrieval rate is calculated as the percentage of images belonging to the same class as the query found in its n top matches. After a number of iterations, the average retrieval rate (ARR) can be computed and used to assess the algorithm performance.

7.2.3 Retrieval algorithm

The outline of the proposed texture-based retrieval algorithm for VHR optical images can be found in Algorithm 1.

As observed from the algorithm, we propose that our keypoints are also the local maximum pixels extracted by a window size ω_2 . It should be noted that ω_2 can be set the same as or different from ω_1 . We usually set $\omega_2 \geq \omega_1$ to get a coarser density of keypoints and speed up the calculation time. This principle has been employed in previous chapter of the thesis (cf. Chapter 4 and Chapter 6). More details about the sensitivity of the algorithm to ω_2 will be discussed later in Section 7.3.4.

Algorithm 1: Proposed retrieval algorithm

Data: Texture database (N images, N_c classes).
Result: Average retrieval rate (ARR), per-class retrieval rate (RR_c).

begin

- parameter setting;
- load database;
- for** $i = 1$ to N **do**
 - load the query image I_i ;
 - compute gradient magnitude and orientation of I_i ;
 - extract 2 local extrema sets $S_{\omega_1}^{\max}(I_i)$ and $S_{\omega_1}^{\min}(I_i)$;
 - extract the keypoint set $S_i = S_{\omega_2}^{\max}(I_i)$;
 - for** $p \in S_i$ **do**
 - extract $\delta^{\text{LED}}(p)$ using (7.10);
 - end**
 - consider the LED feature point cloud $F_i \leftarrow \{\delta^{\text{LED}}(p)\}_{p \in S_i}$;
 - estimate the feature mean vector μ_i of F_i as in (7.12);
 - estimate the feature covariance matrix C_i of F_i as in (7.13);
- end**
- form the distance matrix D :
- for** $i = 1$ to N **do**
 - for** $j = i + 1$ to N **do**
 - compute the distance $d(I_i, I_j)$:
 - if** use Mahalanobis **then**
 - $d(I_i, I_j) = d_{\text{mahalanobis}}(I_i, I_j)$ calculated as in (7.14);
 - else**
 - $d(I_i, I_j) = d_{\text{riemannian}}(I_i, I_j)$ calculated as in (7.15);
 - end**
 - end**
 - $D(i, j) = d(I_i, I_j)$;
 - $D(j, i) = d(I_i, I_j)$;
 - $D(i, i) = 0$;
- end**
- use cross-validation technique:
- for** iteration $t = 1$ to T **do**
 - randomly select n images from each class;
 - for** $i = 1$ to $n \times N_c$ **do**
 - find n top matches (i.e. closest distances) to I_i ;
 - compute retrieval rate (RR) for I_i ;
 - end**
 - compute per-class retrieval rate (RR_c^t) at iteration t ;
 - compute average retrieval rate (ARR^t) at iteration t ;
- end**
- compute final mean RR_c ;
- compute final mean ARR ;

end

7.3 Retrieval experiments

7.3.1 Studied sites and data

For the study of vine cultivation in this chapter, we exploit two VHR panchromatic images acquired by the Pléiades satellite, CNES©, with a spatial resolution of 70cm at nadir, resampled at 50cm. These image data were captured from the regions of Pessac-Léognan and Saint-Emilion, both in France (see their locations from Figure 7.4(a)) on August 22, 2012 and September 3, 2013, respectively. In wine-growing region of Bordeaux, the landscape is mainly dominated by vineyards surrounded by typical peri-urban land covers. A mix of urban, vineyard and forest zones is found from the content of these images. Due to the appearance of different types of textures, four main classes are considered including the forest, bare soil, urban zones and vine fields. It is worth noting that within the vineyard class, there are several vine parcels that are planted under different orientations, as well as from different development stages. For retrieval process, a texture database, with non-equivalent number of image patches (128×128 pixels) per class, was created from each of the two images. We note that there exist some confusions among classes, especially between the bare soil and vine classes. There are several damaged or dead vine fields which destroy the aligned structures of vine textures. In addition, vine fields with the row spacing close to the satellite resolution may introduce smooth and mitigated textures, which become similar to bare soils. Related to this point, in Saint-Emilion region, vine row spacing varies from 1.4m to 2m. Hence, the aligned row structures appear quite clearly within vine fields. On the other hand, Pessac-Léognan region involves vine parcels with row spacing close to 1m. The vine textures are thus mitigated and become more homogeneous. Therefore, although there are only four classes from each database, it is still challenging for texture retrieval task.

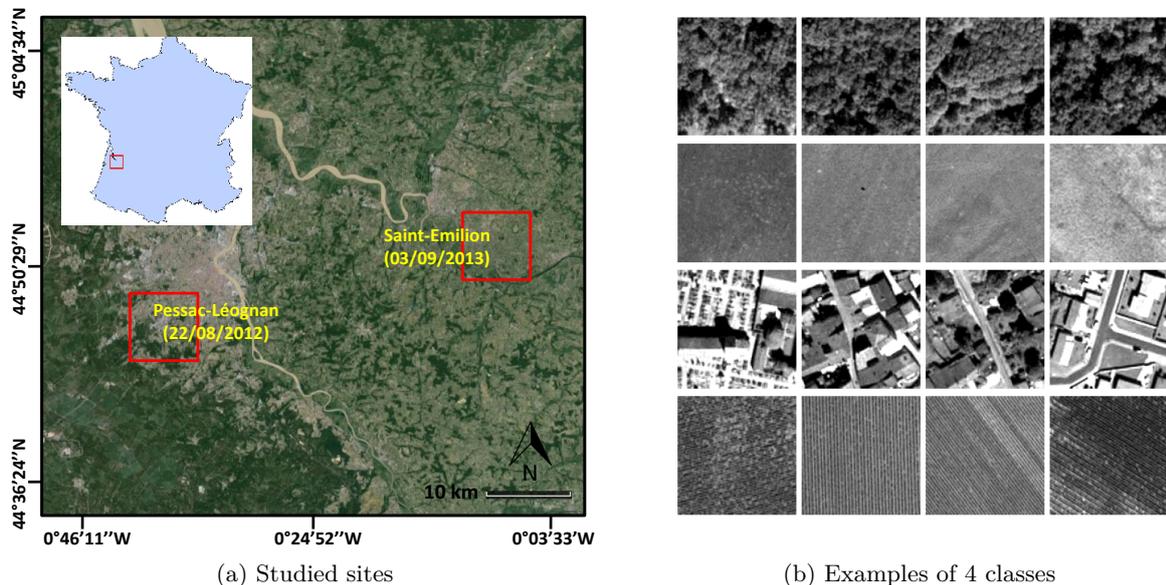


Figure 7.4 – (a) The studied sites of Pessac Léognan and Saint-Emilion regions, France. (b) Examples of 4 texture classes including forest, bare soil, urban zones and vine fields (from top to down) extracted from the *Emilion 03-09-13* database.

For a better explanation, the two databases are named *Pessac 22-08-12* (including 445 patches in total) and *Emilion 03-09-13* (984 patches) in the rest of this chapter. Figure 7.4(b)

illustrates some texture patches of the four classes from the *Emilion 03-09-13* database. The number of image patches per class within each database can be found in Table 7.1.

Table 7.1 – Number of image patches per class within each database for retrieval experiment.

Database	Forest	Bare soil	Urban	Vine fields	Total
<i>Pessac 22-08-12</i>	66	53	147	179	445
<i>Emilion 03-09-13</i>	44	32	27	881	984

7.3.2 Experiment setup

The proposed retrieval framework is applied to the two databases following Algorithm 1. For both data, local max and local min pixels are detected using a 3×3 window ($\omega_1 = 3$), local max keypoints are extracted by the 7×7 window ($\omega_2 = 7$). Here, we set $\omega_2 > \omega_1$ to accelerate the computational time. Experiments show that ω_2 set from 3 to 11 can bring similar retrieval performance. And, the lower ω_2 is (i.e. higher density of keypoints), the more calculation time is required. Next, the number K of closest maxima and closest minima around each keypoint can be set from 10 to 30. We will show later that the results obtained by K equal to 15 and 20 are not significantly different. Moreover, a detailed analysis of the sensitivity of the proposed algorithm to parameters ω_2 and K in terms of ARR and computational time will be provided in Section 7.3.4.

For a comparative study, several reference methods are also implemented including:

- 1) three statistical local texture descriptors: the gray-level cooccurrence matrix (GLCM) [6, 35, 97], the Gabor filter banks (GFB) [12, 98, 99] and the local Weber descriptor (WLD) [92, 101]. The GLCM and GFB appear to be two of the most widely used methods for texture analysis in remote sensing imagery. They have been adopted for vine detection task within the last ten years [35, 97–99]. Meanwhile, the WLD is one of the recent local descriptors in computer vision;
- 2) three distribution models of wavelet coefficients: the multivariate Gaussian model (MGM), the spherically invariant random vectors (SIRV) and the Gaussian copula-based model (GCM) [95, 102]. These methods are the most recent wavelet-based techniques proposed to tackling texture-based retrieval and vine detection tasks. They are considered to give state-of-the-art retrieval performance for our two databases;
- 3) the pointwise (PW) descriptor proposed previously in Chapter 4. This descriptor only exploits the radiometric and spatial information from local extrema points. Gradient features are not considered. For a reminder, the LED can be considered as the improved version of PW by integrating gradient features and taking into account the rotation-invariant property. The comparison to PW descriptor allows us to validate the significant role of gradient features to characterize textural features in this study of vine cultivation.

We note that the implementations of the three model-based techniques (i.e. MGM, SIRV, GCM) are inherited from [95, 102]. Then, the three statistical descriptors (GLCM, GFB, WLD) are implemented using a keypoint-based approach as in Chapter 6 (cf. see Section 6.4.2). They are generated only at keypoint positions to form dedicated feature point clouds, similar to the principle of our strategy. The objective is to perform an equivalent comparison to the proposed LED. Without loss of generality, the window size (W) set to compute these descriptors at

keypoints is varied from 30×30 pixels to 50×50 pixels. Then, the window size which maximizes their performance is adopted. We note that the three wavelet-based techniques can not be generated by such a keypoint-based approach since they densely employ all pixels from the image to perform the wavelet transform.

7.3.3 Results and discussion

Table 7.2 shows the texture retrieval performance on the 2 databases yielded by the proposed algorithm compared to reference methods. Here, the last four rows present four combinations of the proposed LED strategy by setting K equal to 15 or 20, and using the simplified Mahalanobis distance in (7.14) or the Riemannian distance in (7.15). In our implementation, the cross-validation procedure is activated by setting $T = 100$ iterations and $n = 25$ images/class for each iteration. We observe that the proposed approach provides the best ARR for both data sets with an ARR equal to 85.79% for *Pessac 22-08-12* and 89.43% for *Emilion 03-09-13*. In terms of the number K , a slightly better performance is achieved for $K = 20$ than for $K = 15$ but not very significantly. This issue emphasizes the robustness of LED to the number of local extrema considered during its construction. In addition, the Riemannian metric seems to perform more efficiently than the simplified Mahalanobis distance. Hence, we suggest that retrieval process should take into account the Riemannian distance for dissimilarity measurement. On the other hand, the Mahalanobis distance will be in fact exploited in the next stage of supervised classification of vine detection application. We explain this remark in more details in the Section 7.4. Another important remark is that compared to the pointwise (PW) descriptor, the proposed LED has an ARR enhancement of 7.55% for *Pessac-22-08-12* and 6.25% for *Emilion-03-09-13*, with the same parameter setting. This issue confirms the significant role of structural gradient features integrated into LED feature vectors.

Table 7.2 – Retrieval performance on the two VHR texture databases using different methods in terms of average retrieval rate ARR (%).

Method	Pessac 22-08-12	Emilion 03-09-13
MGM	77.51	78.35
SIRV	60.58	60.16
GCM	76.88	75.91
GLCM	54.56	64.57
GFB	61.37	62.39
WLD	64.38	73.88
PW	78.24	83.18
LED (K=15,Mahalanobis)	83.42	87.90
LED (K=20,Mahalanobis)	83.78	88.18
LED (K=15,Riemannian)	85.63	89.01
LED (K=20,Riemannian)	85.79	89.43

Finally, in terms of per-class performance, Table 7.3 shows that the proposed approach can provide good performance for all classes. The per-class retrieval rates for both data sets appear quite homogeneous. In particular, it is very effective in retrieving local structural items such as buildings, structured forests and aligned vine rows. The reason is that we have focused on characterizing local textural features within VHR images and taken them into account during the construction of our proposed LED. Our method reaches the best retrieval rate for urban and vine classes (i.e. 95.97% and 78.34% for *Pessac*; 98.87% and 77.69% for *Emilion*). Within

the 2 databases, there exists a confusion between the bare soil and vine classes. It is caused by some homogeneous or mitigated textures from low-spacing vine fields. Also, some damaged or dead vines destroy the aligned structures of vine rows. That is why the retrieval rates on bare soil and vineyard classes are more limited than for forest and urban classes. As mentioned in Section 7.3.1, from *Pessac* site, most of the vine parcels have low-spacing rows (close to 1m). This introduces smoother vine textures similar to bare soils. The result on bare soil for this site is hence decreased (i.e. 77.75% compared to 85.69% for *Emilion*). Therefore, the total ARR is lower (85.79% compared to 89.43%). Related to the performance of reference approaches, the wavelet-based techniques including MGM and GCM (the first 2 columns) achieve very good performance on forest and bare soils. On the contrary, they yield poor results on urban and vine classes. For example, MGM can produce 96.51% for forest, 95.76% for bare soil, but only 43.99% for vineyard (at *Pessac* database). This reduces its overall ARR (77.51% compared to 85.79% yielded by our method). The same remark can be given for the GCM method. The reason is that the forest zones and bare soils induce very homogeneous textures (i.e. stationary) which are appropriated to the Gaussian models considered by these methods. Meanwhile, in urban zones and vine fields, the notion of local textures and structures is more significant. It would not be relevant to model these two classes by multivariate Gaussian distributions. In conclusion, Table 7.3 shows that the proposed method can provide homogeneous results for all classes. More importantly, it satisfies our first goal of retrieving structural and textural items from the scene, especially vine-plot structures. This issue ensures a good preparation for the next stage of vine parcel detection by a supervised classification scheme.

Table 7.3 – Per-class retrieval accuracy (%) yielded by different methods.

Class/Method	MGM	GCM	GLCM	WLD	PW	LED (K=20)	LED (K=20)
						Mahalanobis (7.14)	Riemannian (7.15)
<i>Pessac 22-08-12</i>							
Forest	96.51	94.54	61.61	88.84	79.08	90.12	92.37
Bare soil	95.76	92.48	56.53	61.76	76.76	74.96	77.75
Urban	73.78	75.28	48.37	57.97	83.00	95.97	94.64
Vineyard	43.99	42.22	51.75	48.94	74.13	74.07	78.34
ARR	77.51	76.88	54.56	64.38	78.24	83.78	85.79
<i>Emilion 03-09-13</i>							
Forest	93.71	84.12	79.20	83.73	90.88	94.35	97.93
Bare soil	99.71	96.25	76.26	78.80	80.99	83.82	85.69
Urban	86.93	87.51	52.84	79.73	88.59	98.87	96.40
Vineyard	33.05	39.75	49.96	53.25	72.28	75.68	77.69
ARR	78.35	75.91	64.57	73.88	83.18	88.18	89.43

7.3.4 Sensitivity to parameters

This subsection aims at studying the sensitivity of the proposed retrieval framework to its parameters. As observed from the full retrieval algorithm (cf. Algorithm 1), three main parameters are required in our framework including: the window size to detect local maximum and local minimum pixels (ω_1), the window size to extract local max keypoints (ω_2) and the number of closest extrema considered for each keypoint to construct its LED (K). Since ω_1 needs to be small enough to ensure sufficiently dense local extrema to support the computation of LED, we propose to fix it to 3×3 pixels in this work. Hence, only two parameters are considered to

influence the performance of our method: ω_2 and K . We now investigate the sensitivity of the algorithm to each of them.

7.3.4.1 Sensitivity to the window size ω_2

Figure 7.5(a) shows the performance of the proposed algorithm obtained by fixing $\omega_1 = 3$, $K = 20$ and varying ω_2 from 3 to 11. Experiments are performed on the *Emilioni 03-09-13* data. We note that using the *Pessac 22-08-12* data provides a similar observation. First of all, a decrease in ARR from 89.99% to 87.32% is observed when ω_2 increases from 3 to 11. This remark can be explained as follows. As previously discussed in Section 7.2.3, the higher ω_2 , the coarser the density of keypoints. When the number of keypoints N_i considered for each image I_i decreases, the estimation accuracy of the feature covariance matrix C_i in Equation (7.13) is reduced. The calculation of Riemannian distance (Equation (7.15)) is thus influenced. In other words, the more keypoints we use to characterize the image, the more precisely covariance matrix is estimated, and hence the higher ARR is obtained. Nevertheless, since only a decrease of 1.03% (from 89.99% to 88.96%) is yielded when ω_2 switches from 3 to 9, the method can be considered to be less sensitive to parameter ω_2 . Next, in terms of computational cost, the LED feature extraction time per image patch is significantly reduced when ω_2 increases. It is also because the fewer number of keypoints is exploited. Hence, we can conclude that ω_2 involves a compromise between the accuracy of covariance matrix estimation and the rapidity of LED feature extraction. Thus, it leads to a compromise between the retrieval rate and the calculation time of our algorithm. To this end, although the best performance in ARR (89.99%) is obtained by setting $\omega_2 = 3$, one may prefer setting it to 5 or 7 to speed up the computation time (i.e. as our parameter setting with $\omega_2 = 7$ in Section 7.3.3). Indeed, by setting $\omega_2 = 7$, we gain about 55% of time (i.e. reduced from 0.886s to 0.396s per image). However, only a reduction of 0.56% in ARR (i.e. from 89.99% to 89.43%) is resulted.

7.3.4.2 Sensitivity to the number K

Similarly, the algorithm sensitivity to the parameter K (i.e. the number of closest extrema considered for each keypoint for the generation of its LED feature vector) can be found in Figure 7.5(b). Here, we fix $\omega_1 = 3$, $\omega_2 = 7$ and vary K from 10 to 30. Firstly, a stable performance can be observed with an ARR varying from 88.6% to 89.43%. The variation involves 2 stages. When K increases from 10 to 20, ARR is enhanced to reach the highest value (89.43%). Then, if K continues to increase from 20 to 30, ARR starts to be reduced. Here is our explanation. The parameter K in our strategy plays a similar role to that of the sliding window $W \times W$ set for the construction of classical dense descriptors. Hence, it has a similar behavior. At the first stage, when K increases, more information of the local neighborhood (i.e. which includes K closest maxima and K closest minima) around the keypoint are taken in account. Thus, the performance of LED is improved. If we continue to increase the value of K , the equivalent support neighborhood size becomes larger and larger. Although more information is exploited, we may lose the notion of local feature and signal stationarity. This reduces the capacity of LED to discriminate local structures and textures. Hence, the retrieval performance is decreased. In terms of computational time, the more the number of extrema considered for each keypoint, the more the calculation time. In general, at $K = 20$, only a total time of 389.65 seconds is necessary for the complete algorithm to produce 89.43% retrieval accuracy for 984 patches of the *Emilioni 03-09-13* database. This issue makes the proposed strategy very effective and competitive in both retrieval performance and computational cost. In conclusion, the two figures 7.5(a) and 7.5(b) show that the proposed LED method is not very sensitive to its parameters.

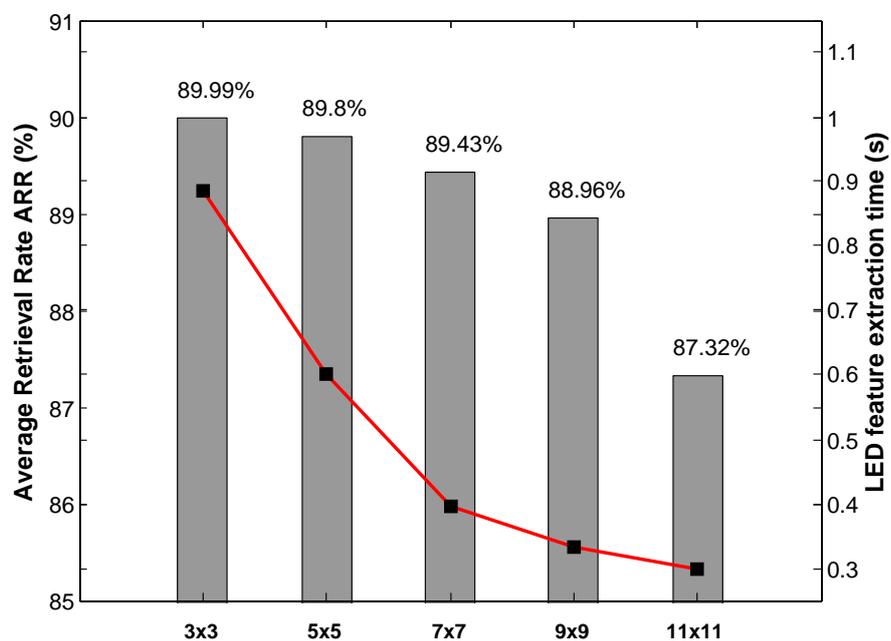
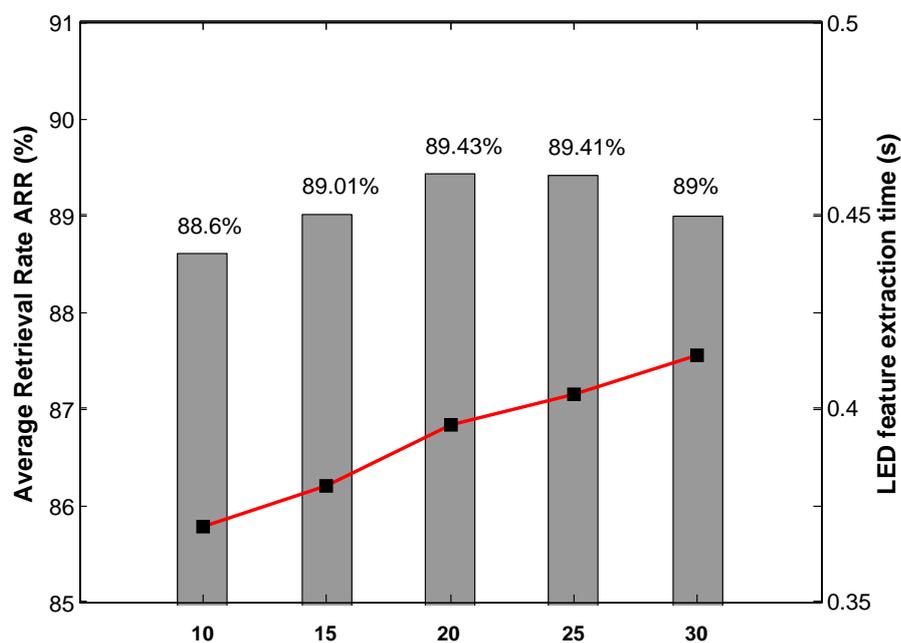
(a) Sensitivity to window size ω_2 for keypoint extraction.(b) Sensitivity to the number K of closest extrema for LED construction.

Figure 7.5 – Sensitivity of the proposed retrieval framework to its parameters in terms of average retrieval rate (%) and LED feature extraction time (s). Experiments are performed on *Emilion 03-09-13* database using the Riemannian distance (7.15).

A stable performance can be adopted with a wide range of parameters: $\omega_2 \in \{3, \dots, 9\}$ and $K \in \{10, \dots, 30\}$.

7.4 Application to vineyard detection based on supervised classification

7.4.1 Supervised classification algorithm

In the previous section, the proposed LED has proved its capacity to characterize local structural and textural features from VHR images. Its performance has been validated for texture-based image retrieval system with very competitive results. We now tackle the main purpose of vineyard parcel detection using VHR satellite images. As mentioned in the introduction section of the chapter, we would like to perform a supervised classification algorithm to distinguish vine parcels from others classes present from the image content such as forest zones, bare soils and urban areas. Here, only keypoints are exploited for classification, not all pixels from the image (like in Chapter 4 and Chapter 6). In details, the proposed supervised classification algorithm is activated as in Algorithm 2 with the helps of the k-nearest neighbor (kNN) classifier [103] and the Mahalanobis distance. Here, the kNN is chosen thanks to its effective performance as well as its simplicity during our experimentation. Nevertheless, other classifiers such as the Minimum Distance (i.e. exploited in Chapter 6) or the Support Vector Machine (SVM), etc. can be also applied to this task.

In Algorithm 2, the computation of the distance between each keypoint $p \in S$ and each image patch I_i in the training set, denoted by $d(p, I_i)$, is defined as follows:

$$\begin{aligned} d(p, I_i) &= d_{\text{Mahalanobis}}(\delta^{\text{LED}}(p), F_i) \\ &= \left(\delta^{\text{LED}}(p) - \mu_i \right) C_i^{-1} \left(\delta^{\text{LED}}(p) - \mu_i \right)^T, \end{aligned} \quad (7.16)$$

where μ_i and C_i are the mean feature vector and the feature covariance matrix estimated for the point cloud F_i as in Equations (7.12) and (7.13).

We note that the Mahalanobis distance is exploited here since it can be measured from one sample feature point $\delta^{\text{LED}}(p)$ to the point cloud F_i . On the other hand, the Riemannian metric is not applicable since it is a distance between two point clouds whose covariance matrices are both required for its computation as in (7.15). Hence, it is more relevant for texture retrieval task when the distance between two image patches needs to be computed.

Algorithm 2: Proposed supervised classification algorithm

Data: Input image I , training data set of N images.
Result: Classification result of I .

begin

- parameter setting;
- load training data set;
- for** $i = 1$ to N **do**
 - load the image I_i ;
 - compute gradient magnitude and orientation of I_i ;
 - extract 2 local extrema sets $S_{\omega_1}^{\max}(I_i)$ and $S_{\omega_1}^{\min}(I_i)$;
 - extract the keypoint set $S_i = S_{\omega_2}^{\max}(I_i)$;
 - for** $p \in S_i$ **do**
 - extract $\delta^{\text{LED}}(p)$ using (7.10);
 - end**
 - consider the LED feature point cloud $F_i \leftarrow \{\delta^{\text{LED}}(p)\}_{p \in S_i}$;
 - estimate the feature mean vector μ_i of F_i as in (7.12);
 - estimate the feature covariance matrix C_i of F_i as in (7.13);
- end**
- load the image I ;
- compute gradient magnitude and orientation of I ;
- extract 2 local extrema sets $S_{\omega_1}^{\max}(I)$ and $S_{\omega_1}^{\min}(I)$;
- extract the keypoint set $S = S_{\omega_2}^{\max}(I)$;
- for** $p \in S$ **do**
 - extract $\delta^{\text{LED}}(p)$ using (7.10);
 - for** $i = 1$ to N **do**
 - compute the distance $d(p, I_i)$ as in (7.16);
 - end**
 - find k nearest neighbors corresponding to k closest distances;
 - affect the major class present from the k nearest neighbors to p ;
- end**

end

7.4.2 Experimental results

This subsection describes our experimental setup and provides classification and detection results yielded by the proposed strategy compared to several reference methods. It should be noted that our main objective is the detection of vine parcels within the image, not the classification precision of all classes obtained from the supervised classification algorithm. Hence, we are interested in a detection problem with only two primary classes: *vine* and *non-vine* (i.e. which consist of all remaining classes including forest zones, urban/man-made areas, and bare soils.). For the performance evaluation of the proposed method compared to reference methods, both qualitative and quantitative assessments are performed. We propose to evaluate the number of good detections (GD), false alarms (FA) and missed detections (MD) as well as to calculate the percentage of total errors (P_{TE}) and the percentage of overall accuracy P_{OA} . More details about the computation of these indexes can be consulted in Appendix A.

The proposed classification algorithm (Algorithm 2) is applied to a crop of 2000×1700 pixels extracted from the VHR panchromatic image acquired in Saint-Emilion region. The image crop

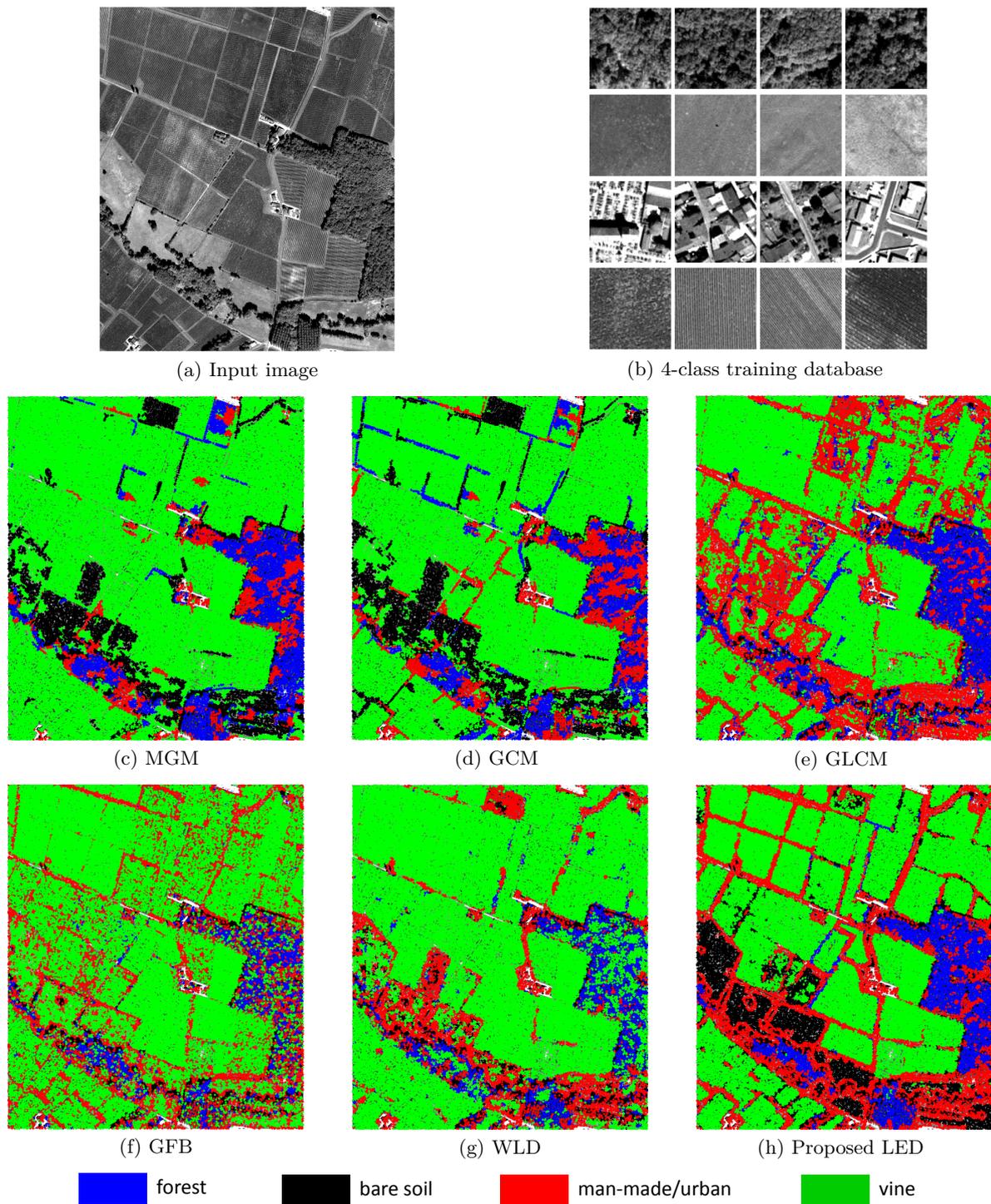


Figure 7.6 – Supervised classification results for Saint-Emilion site yielded by the proposed algorithm compared to reference methods. (a) Input image (2000×1700 pixels). (b) Examples of training patches from the *Emilion 03-09-13* database including 4 classes: forest, bare soil, urban and vine fields. From (c) to (h) Classification results yielded by the multivariate Gaussian model (MGM), Gaussian copula model (GCM), Gray-level cooccurrence matrix (GLCM), Gabor filter bank (GFB), Weber local descriptor (WLD) and the proposed local extrema-based descriptor (LED) methods. Colored labels affected for the classes of forest, bare soil, man-made/urban and vineyard are blue, black, red and green, respectively.

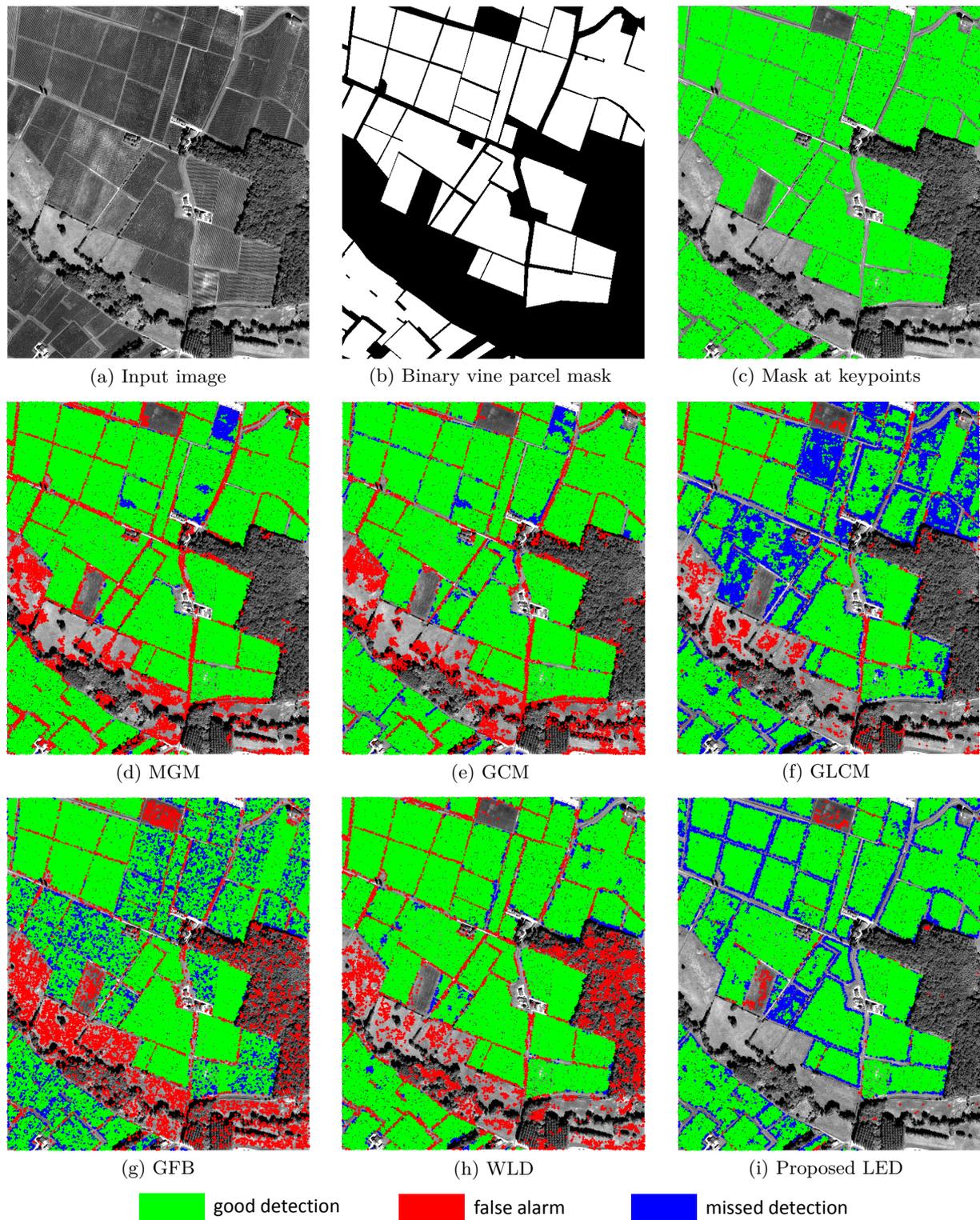


Figure 7.7 – Vine parcel detection results for Saint-Emilion site yielded by the proposed algorithm compared to reference methods. (a) Input image (2000×1700 pixels). (b) Binary map consisting of *vine* fields in white and *non-vine* fields in black. (c) Ground truth detection results at keypoints with *vine* points in green. From (d) to (i) Detection results yielded by the multivariate Gaussian model (MGM), Gaussian copula model (GCM), Gray-level cooccurrence matrix (GLCM), Gabor filter bank (GFB), Weber local descriptor (WLD) and the proposed local extrema-based descriptor (LED) methods. Colored labels affected for Good detection (GD), False alarm (FA) and Missed detection (MD) points are green, red, blue, respectively.

is shown in Figure 7.6(a). As for the retrieval experimental study in Section 7.3.3, similar parameters are set for the generation of LED feature vectors: $\omega_1 = 3$, $\omega_2 = 7$ and $K = 30$. Then, the kNN classification is performed by setting $k = 10$ nearest neighbors. The *Emilion 03-09-13* database with 984 image patches (see Table 7.1) is exploited as training set. A total time of about 30 minutes is taken by the full algorithm using basic MATLAB implementation on a machine *Xeon 3.6GHz, 16GB RAM*.

Figures 7.6 and 7.7 provide the supervised classification and vine detection results yielded by the proposed LED compared to several reference methods. Here, we show the results produced by two wavelet-based techniques: the multivariate Gaussian model (MGM) and the Gaussian copula model [95, 102]; and three classical statistical methods: the Gray-level cooccurrence matrix (GLCM) [6], the Gabor filter banks (GFB) [12] and the Weber local descriptors (WLD) [92]. The implementations of these reference methods are carried out similarly to the previous retrieval task (i.e. similar to the implementations in Chapter 6). Supervised classification procedure is performed with parameters dedicated to each method to ensure an equivalent comparison. As observed in Figure 7.6, the proposed LED strategy produces well-separated classes from which vineyard parcels are discriminated from other classes (Figure 7.6(h)). Its classification process takes into account the boundaries (i.e. which are in fact dirt roads or ridges) between vine parcels and considers them to be similar as urban/man-made items (marked in red). On the other hand, the two wavelet-based techniques (Figures 7.6(c)-(d)) and the WLD (Figure 7.6(g)) over-smooth these boundaries and provide quite homogeneous vine fields. Hence, they create over-detection results for vineyard class (i.e. most ridges are detected as vines). In terms of scene interpretation, we believe that it is better to classify these roads as man-made items than to detect them as vines. The result in Figure 7.6(h) enables us to recognize different vine parcels. Meanwhile, by over-detecting those ridges as vines, a lot of false alarms are resulted. Hence, the detection performance will be reduced. The two other approaches including the GLCM and the GFB give poorer classification results in which some classes are mixed and vine fields are not well distinguished. Since we do not have a precise classification ground truth for all classes, only qualitative assessment for scene interpretation can be performed from Figure 7.6. Meanwhile, for vine detection task, a full qualitative and quantitative comparison will be provided.

In Figure 7.7, vine detection results are shown in which good detection (GD) points are marked in green while false alarm (FA) and missed detection (MD) points are in red and blue, respectively. Again, the results obtained from GLCM (Figure 7.7(f)) and GFB (Figure 7.7(g)) are not sufficiently good with a great number of FA and MD points. The other results involve a compromise between the number of FA points and the number of MD points. The two wavelet-based approaches (i.e. MGM, GCM) and the WLD generate a lot FA points because of their over-detection problem. On the contrary, the proposed method produces more MD points. However, as observed from the figure, most of the MD points yielded by the proposed LED mostly come from boundary regions (i.e. dirt roads, rides between vine parcels). It is because during the classification process, our method well detect these boundaries but with a larger width, hence resulting some missed vine points. In order to evaluate the performance of the proposed approach compared to the others, Table 7.4 provides some evaluation indicators described in the previous subsection. As observed from the table, the proposed algorithm provides the best detection result in terms of the ratio between good and bad detection points ($GD/FA+MD = 5.4971$) as well as the percentage of total error ($P_{TE} = 10.26\%$) and the percentage of overall accuracy ($P_{OA} = 89.74\%$). An enhancement of 2.16% is achieved compared to the best state-of-the-art method, the GCM with 87.58%. Also, compared to the PW descriptor, the proposed LED improves 2.22% of detection performance. Its again emphasizes the significant role of structural gradient features integrated into the descriptor.

Table 7.4 – Comparison of the numbers of False Alarms (FA), Missed Detections (MD), Good Detections (GD), percentage of Total Errors (P_{TE}) and percentage of Overall Accuracy (P_{OA}) produced by different methods. Experiments are performed on an image crop of 2000×1700 pixels acquired from Saint-Emilion region with $N_{\text{vine}} = 32853$ points, $N_{\text{non-vine}} = 17026$ points.

Method	FA (points)	MD (points)	GD (points)	$GD / (FA+MD)$	P_{TE} (%)	P_{OA} (%)
MGM	6011	676	32177	4.8119	13.41	86.59
GCM	5160	1035	31818	5.1361	12.42	87.58
GLCM	2469	8200	24653	2.3107	21.39	78.61
GFB	6316	5159	27694	2.4134	23.01	76.99
WLD	6026	858	31995	4.6477	13.80	86.20
PW	667	5557	26696	4.2892	12.48	87.52
Proposed LED	412	4708	28145	5.4971	10.26	89.74

Table 7.5 – Comparison of the numbers of False Alarms (FA), Missed Detections (MD), Good Detections (GD), percentage of Total Errors (P_{TE}) and percentage of Overall Accuracy (P_{OA}) produced by different methods. Experiments are performed on an image crop of 1300×1700 pixels acquired from Pessac-Léognan region with $N_{\text{vine}} = 18176$ points, $N_{\text{non-vine}} = 10337$ points.

Method	FA (points)	MD (points)	GD (points)	$GD / (FA+MD)$	P_{TE} (%)	P_{OA} (%)
MGM	2714	1496	16230	3.4828	16.34	83.66
GCM	2309	1408	16768	4.5112	13.04	86.96
GLCM	2010	4589	13587	2.0589	23.14	76.86
GFB	2224	5749	12427	1.5586	27.96	72.04
WLD	3704	1836	16340	2.9495	19.43	80.57
PW	618	4699	13477	2.5347	18.65	81.35
Proposed LED	1270	1688	16488	5.5740	10.37	89.63

Last but not least, a similar classification framework is applied to a crop of 1300×1700 pixels extracted from the acquired image in Pessac-Léognan region. The corresponding *Pessac 22-08-12* database is employed as training set. Figure 7.8 and Table 7.5 provide the detection results yielded by the proposed LED compared to reference methods. From the figure, we observe that this site involves more urban zones. More significantly, vine fields here appear more homogeneous since their row spacing is small and close to the sensor resolution. For a reminder, the Pléiades satellite’s spatial resolution at nadir is 70cm. The distance between vine rows in Pessac-Léognan site is from 1m to 1.2m. In Saint-Emilion region, this row spacing varies from 1.4m to 2m. Despite this challenge, the detection performance of all methods is quite similar to the previous case of Saint-Emilion region. We again observe the best performance from the proposed LED with $GD / FA+MD = 5.574$, $P_{TE} = 10.37\%$ and $P_{OA} = 89.63\%$. An improvement of 2.67% is made compared to the best reference method (GCM with $P_{OA} = 86.96\%$). In conclusion, the proposed method provides efficient and superior performance compared against all the reference methods mentioned in our experimentation. It is evaluated and validated for both the texture retrieval task and the application of vine detection task. Its robustness is proved for both studied sites with different characteristics from the vine parcels.

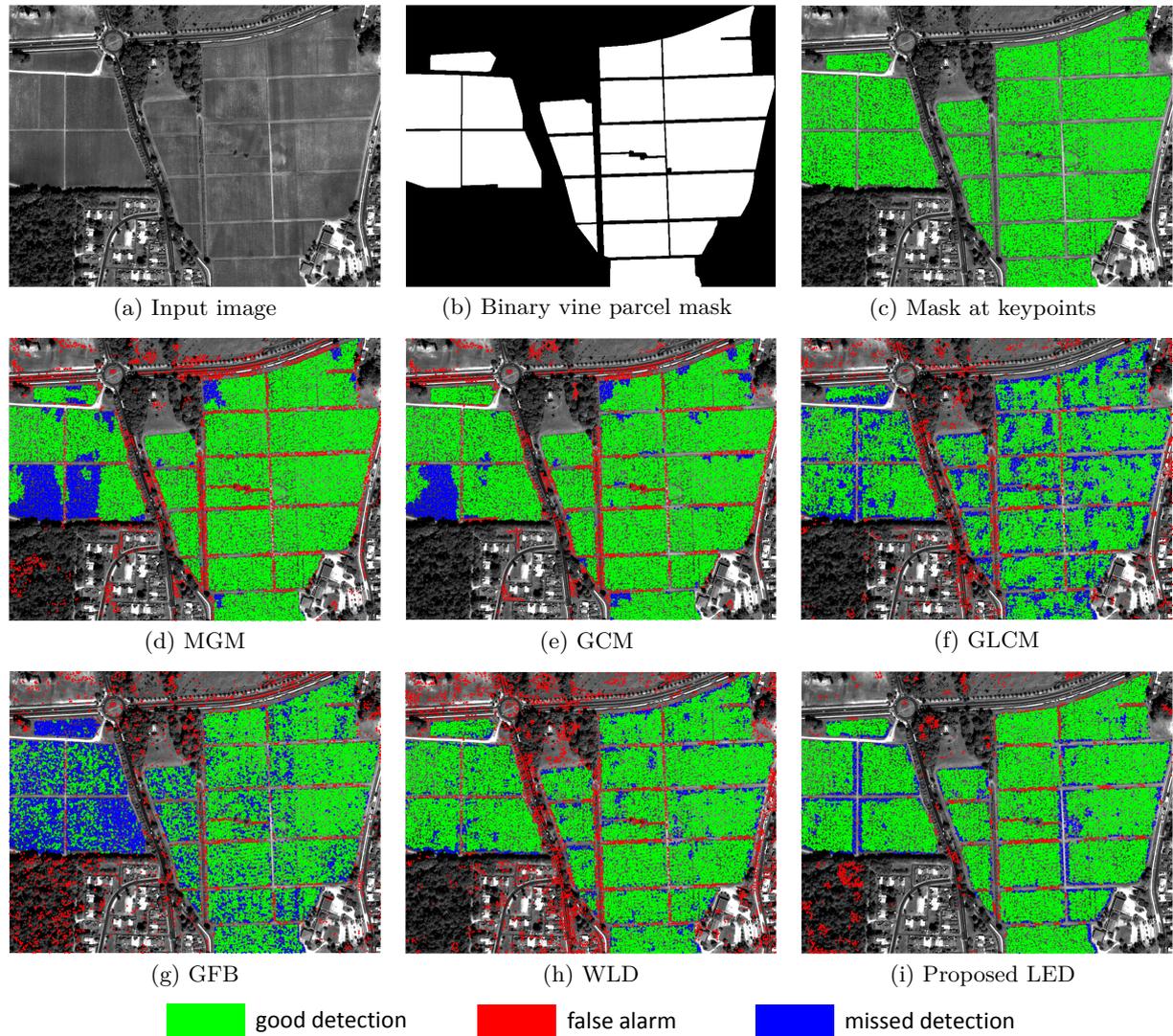


Figure 7.8 – Vine parcel detection results for Pessac-Péognan site yielded by the proposed algorithm compared to reference methods. (a) Input image (1300×1700 pixels). (b) Binary map consisting of *vine* fields in white and *non-vine* fields in black. (c) Ground truth detection results at keypoints with *vine* points in green. From (d) to (i) Detection results yielded by the multivariate Gaussian model (MGM), Gaussian copula model (GCM), Gray-level cooccurrence matrix (GLCM), Gabor filter bank (GFB), Weber local descriptor (WLD) and the proposed local extrema-based descriptor (LED) methods. Colored labels affected for Good detection (GD), False alarm (FA) and Missed detection (MD) points are green, red, blue, respectively.

7.5 Conclusion

In this chapter, texture retrieval from VHR imagery applied to the detection of vineyard parcels has been investigated. The local extrema descriptor (LED) has been developed to characterize textural and structural features from the image content. This novel descriptor is an improved version of the pointwise (PW) descriptor proposed previously in Chapter 4. It is constructed by incorporating the radiometric, geometric and structural gradient features extracted from the local extrema (i.e. local maxima and local minima) around each keypoint. The performance of LED has been proved and validated for both retrieval and supervised classification tasks using VHR panchromatic Pléiades image data. One can observe that the proposed LED is easy to build and implement, feasible to improve or extend and effective for VHR images when dealing with textural and structural items from the image. In addition, structural information appears to be an important parameter which needs to be encoded in the scope of vine cultivation.

To this end, the chapter again confirms the effectiveness and relevance of the pointwise approach for texture analysis in VHR imagery. We now move on to the final chapter where some conclusions are drawn from the thesis. Then, based on the remaining issues dedicated to each chapter, we discuss several research studies which can be potentially investigated in future work.

Part IV

Conclusion

Conclusions and Perspectives

We can't plan life. All we can do is be available for it.

Lauryn Hill

8.1 Conclusion	143
8.2 Perspectives	144

8.1 Conclusion

This thesis work has proposed a novel pointwise approach for texture analysis in the scope of remote sensing imagery. This strategy exploits only characteristic pixels to represent and characterize textural features. To perform such an approach, the local maximum and local minimum pixels have been proved to be relevant and efficient thanks to their capacity of capturing significant information from the image content.

During this work, we have developed several frameworks and applications based on texture analysis using the pointwise approach. The first idea of encoding the inter-connection of keypoints was successfully performed in Part II by using the weighted graph model. Signal processing on graphs was studied to provide different graph's characteristics and transformations relevant for texture analysis. Our investigation pointed out that the graph spectral domain is relevant to perform multiscale texture representation and description, while the vertex domain is promising to perform tracking and detection tasks based on texture coherence. As a result, pointwise graph-based texture descriptors including the PW+SGC (i.e. pointwise descriptor + spectral graph clustering) and the PW+SGWT (i.e. pointwise descriptor + spectral graph wavelet transform) were generated in Chapter 4 to perform texture-based classification task applied to VHR optical images. Their performance was evaluated and confirmed using both panchromatic and multispectral Pléiades images. Then, in Chapter 5, a graph-based measure of texture coherence was proposed to tackle texture-based detection task from SAR imagery. This measure provided an effective performance when tackling change detection task using RADARSAT image data. Moreover, some promising early results were obtained when the proposed strategy was applied to the detection of glacier flows using TerraSAR-X images.

Part III continued to develop the proposed pointwise approach without using weighted graph structures due to their high computational complexity. The two novel local descriptors including the PW-COG (i.e. pointwise covariance matrix of oriented gradients) and LED (i.e. local extrema descriptor) were proposed in Chapters 6 and 7, respectively. Along with radiometric

and geometric information, structural properties given by gradient features were also integrated into these descriptors in order to deal with oriented texture zones observed from the image. By exploiting the relevant Riemannian distance metric, their performance in terms of texture characterization and discrimination was confirmed by performing texture-based retrieval and classification tasks using panchromatic Pléiades images. Furthermore, both descriptors involve a fast implementation and they are easy to be improved or extended by adding other features during their construction.

The novelty of this work is no doubt is use of local extrema pixels for non-dense representation and characterization of textures. However, its contributions rely significantly on the fact that several techniques and methods (i.e. graph theory, covariance approach, geometric distance measurement, etc.) were investigated to perform and combine with the pointwise approach. Also, various applications such as texture-based segmentation, classification, retrieval, change and displacement detection, etc. using remote sensing images (both optical and SAR data) were tackled. To this end, since the thesis's application domain is wide, several perspective work can be potentially investigated in the future. We discuss this issue in the next section.

8.2 Perspectives

The following questions and suggestions can be considered in future research studies:

1. Throughout this thesis work, several local texture descriptors (i.e. PW, PW-COG, LED) were developed for VHR optical images using the pointwise approach (cf. Chapters 4, 6 and 7). The question is how we can adapt and extend them for SAR images. The local maximum and local minimum pixels were proved to be relevant to represent and capture textural information from optical images. However, in SAR imaging, their appearance is no longer reliable due to the presence of speckle noise. In Chapter 5, a smoothing operator using lowpass filter was performed before the extraction of local extrema points. Hence, is it better to continue this strategy or to redefine other types of characteristic points more adapted to SAR imagery? Also, it is worth noting that radiometric and structural features from SAR images are perturbed by speckle. Therefore, future studies should adapt their computation during the construction of SAR texture descriptors.
2. In our work, the above descriptors were generated only on keypoints (which were also the local maximum pixels). One may wonder about the possibility to perform a dense approach for all pixels from an image. We note that our descriptors including PW, PW-COG and LED can be generated for all image's pixels. Nevertheless, for an input image, it would be interesting to produce an image of texture measures given by these descriptors (like the Haralick measures of contrast, energy, homogeneity, etc. given by the GLCM technique [6]). This idea can be considered for further investigation.
3. In Chapter 4, to deal with VHR multispectral images, we proposed to insert complementary spectral information into the pointwise descriptor. Our method was to perform the principal component analysis (PCA) on the input image and then use only the first and second PCA bands (cf. Section 4.3.1). The question is whether this PCA-based approach is the best strategy to incorporate spectral and spatial information when dealing with multispectral or especially hyperspectral images. Beside that, when structural gradient features are taken into account (e.g. in PW-COG and LED vectors), is it enough to extract them only on one band, on the first PCA band or is it necessary to encode the multi-band tensor structures (e.g. Di Zenzo gradients [104])?

4. The proposed pointwise graph-based measure of texture coherence in Chapter 5 provided a good performance on change detection between two SAR images. A potential work could be to investigate the strategy on multi-date SAR data to detect major changes occurring within the development of agricultural crops, forests or urban zones over time.
5. Also in Chapter 5, when applying the proposed texture tracking approach to the detection of glacier displacements (cf. Section 5.4), we observed its effective performance on crevasse regions with strong textural properties. However, some limitations were observed on homogeneous and flat regions. Hence, further work is expected to improve the performance of this measure on flat regions.
6. In Chapter 7, the proposed LED was applied to image retrieval framework using Pléiades image database. It is promising to investigate its performance on natural texture databases in computer vision domain. By combining color and texture features into LED vector, our attempt has provided very competitive results on the popular color texture databases such as Vistex [105], STex [106] and Brodatz [107]. Therefore, further research could investigate the proposed descriptors like LED or PW-COG for other applications in image processing and computer vision such as image categorization, matching, object recognition, target detection, etc.
7. Finally, as discussed in Chapter 2, the density of local extrema pixels detected from an image depends on the search window size. By varying this window size, one can perform a multiscale texture analysis using pointwise approach. This method may allow us to potentially improve the performance of all the proposed descriptors and graph-based measures in future investigation.

List of Figures

1	Représentation des textures par l'approche ponctuelle utilisant des maxima et minima locaux	3
2	Représentation des max et min locaux dans des textures locales différentes	4
3	Distribution des points clés dans une image.	5
4	Construction d'un graphe à partir d'un ensemble de points d'intérêt.	5
5	Vecteur de description proposé pour l'analyse ponctuelle de la texture.	6
6	Résultats de classification de la base n° 1.	8
7	Images Radarsat-1 acquises avant et après l'éruption du volcan Nyiragongo en janvier 2002.	10
8	Mesure de changements appliquée aux images avant/après de Goma.	11
9	Courbe de performance des mesures de changement présentées à la figure 8.	12
10	Différents modes de gestion du voisinage et de la similarité entre les situations locales.	12
11	Principe général de corrélation de texture par approche ponctuelle.	13
12	Mesure de déplacement des glacier.	14
13	Résultats de classification non supervisée	15
14	Information géométrique et gradient local des points clés.	16
15	Détection des parcelles viticoles dans le domaine de Saint Emilion.	19
1.1	Example of VHR optical satellites.	23
1.2	Example of VHR optical image acquired by Pléiades-1A satellite.	24
2.1	Example of the first iteration to calculate the first IMF using the EMD method.	30
2.2	Texture representation and discrimination using the local maximum and local minimum pixels within a pointwise approach.	31
2.3	Detection of local maximum and local minimum pixels from an image patch.	32
2.4	Spatial distribution and arrangement of local maximum and local minimum pixels within different texture zones from a studied ROI.	32
2.5	Local max/min detection for two texture patches (140×140 pixels) with 3 different search window sizes.	34
2.6	Distribution of keypoints on the image plane.	35
3.1	Example of a weighted, undirected and unlooped graph.	40
3.2	A signal f on graph \mathcal{G} . Each vertex v_i encapsulates a signal sample $f(i)$	42
3.3	A ring graph.	43
3.4	Example of multi-scale image decomposition using wavelet transform on a local graph.	45
3.5	Low-pass and band-pass filters for SGWT image decomposition in Figure 3.4.	45
3.6	Example of image denoising using wavelet transform on a non-local graph.	46
3.7	Graph construction for keypoints.	47

4.1	Generation of the proposed pointwise descriptor for an understudied pixel.	51
4.2	Proposed framework for VHR panchromatic/multispectral image texture characterization applied to classification task using a pointwise graph-based approach. .	54
4.3	Classification results for Dataset 1.	63
4.4	Classification results for Dataset 2.	64
5.1	Proposed framework for pointwise graph-based SAR image change detection. . .	72
5.2	Appearance of local max and min keypoints on SAR signal.	73
5.3	Appearance of keypoints extracted from a SAR image.	74
5.4	Experimental study for Data set 1.	78
5.5	ROC plots for the proposed change detection method compared to four reference methods.	79
5.6	Experimental study for Data set 2.	82
5.7	Zoomed results for the <i>Red</i> studied crop of Data set 2.	83
5.8	Zoomed results for the <i>Green</i> studied crop of Data set 2.	83
5.9	Comparison of the dense and pointwise LRD methods (with different window sizes (ω) for keypoint extraction) for change detection.	84
5.10	ROC plots for the proposed method within the variation of its parameters. . . .	85
5.11	Support neighborhood considered for measuring the change level at each pixel. .	88
5.12	ROC plots for the proposed change detection method compared to four reference methods using the local min keypoints.	89
5.13	Outline of the proposed graph-based texture tracking for glacier flow detection. .	90
5.14	Input TerraSAR-X images acquired from the Argentiere Glacier.	92
5.15	Glacier displacement result yielded by the proposed algorithm.	93
6.1	Generation of dense COG from the neighborhood $\mathcal{N}_W(p)$ and pointwise COG from the local maximum set $\mathcal{N}_W^{\max}(p)$	101
6.2	Proposed framework for texture-based classification of VHR panchromatic images using the proposed PW-COG descriptor.	102
6.3	Unsupervised classification results for <i>Brodatz</i> data.	108
6.4	Unsupervised classification results for <i>ROI-1</i>	110
6.5	Unsupervised classification results for <i>ROI-2</i>	112
6.6	Supervised classification experiment for <i>ROI-3</i>	113
6.7	Supervised classification results for the <i>Red</i> and <i>Yellow</i> cuts from <i>ROI-3</i>	115
7.1	A general content-based image retrieval framework.	121
7.2	Generation of local extrema descriptors (LEDs) for each query image.	122
7.3	Geometric and gradient information from a local maximum or local minimum pixel. .	123
7.4	Studied sites and examples of texture database.	127
7.5	Sensitivity of the proposed retrieval framework to its parameters.	132
7.6	Supervised classification results for Saint-Emilion site yielded by the proposed algorithm compared to reference methods.	135
7.7	Vine parcel detection results for Saint-Emilion site yielded by the proposed algorithm compared to reference methods.	136
7.8	Vine parcel detection results for Pessac-Péognan site yielded by the proposed algorithm compared to reference methods.	139

List of Tables

1	Nombre de patches utilisés pour la reconnaissance supervisée de type d'utilisation des sols.	18
2	Performance de classification supervisée, évaluée en taux de reconnaissance moyen (%).	18
3.1	Intervention of graph signal processing in Chapter 4 and Chapter 5.	48
4.1	Parameter setting in our implementation. Classification results are shown in Figures 4.3 and 4.4.	61
4.2	Comparison of total computational time (TCT), overall classification accuracy (OCA) and <i>Kappa</i> coefficient for Dataset 1.	62
4.3	Comparison of total computational time (TCT), overall classification accuracy (OCA) and <i>Kappa</i> coefficient for Dataset 2.	65
4.4	Detailed time consumption of the proposed framework.	66
4.5	Computational time for graph construction using different search window sizes for vertex extraction.	66
4.6	Sensitivity of the proposed method to the number of closest extrema (K) used for vertex description.	67
5.1	Change detection performance on data set 1.	80
5.2	Comparison of computational complexity and detection performance from different methods.	88
6.1	Parameter setting in our implementation for unsupervised classification.	105
6.2	Comparison of descriptor computation time, overall classification accuracy (OCA) and <i>Kappa</i> coefficient of different methods to perform unsupervised classification on <i>Brodatz</i> , <i>ROI-1</i> and <i>ROI-2</i> data.	107
6.3	Comparison of computation time and classification accuracy of different methods to perform supervised classification on <i>ROI-3</i> data.	114
6.4	Sensitivity of the proposed method to the window size ω_1	117
6.5	Sensitivity of the proposed method to the window size ω_2	117
6.6	Sensitivity of the proposed method to the window size W	117
6.7	Comparison of overall classification accuracy (OCA) yielded by the DCOG and PW-COG methods by using Euclidean and Riemannian distance metrics.	118
7.1	Number of image patches per class within each database for retrieval experiment.	128
7.2	Retrieval performance on the two VHR texture databases using different methods in terms of average retrieval rate ARR (%).	129
7.3	Per-class retrieval accuracy (%) yielded by different methods.	130
7.4	Vineyard detection performance on Saint-Emilion region.	138
7.5	Vineyard detection performance on Pessac-Léognan region.	138

Bibliography

- [1] M. Tuceryan and A. K. Jain. Texture analysis. *Handbook of pattern recognition and computer vision*, 2:207–248, 1993.
- [2] T. R. Reed and J. M. H. du Buf. A review of recent texture segmentation and feature extraction techniques. In *GVGIP: Image Understanding*, pages 359–372, 1993.
- [3] J. Zhang and T. Tan. Brief review of invariant texture analysis methods. In *Journal of Pattern Recognition* 35, pages 735–747, March 2001.
- [4] L. Ma and J. Yu. Texture segmentation based on local feature histograms. In *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pages 3349–3352, 2011.
- [5] H.-C. Lin, L.-L. Wang, and S.-N. Yang. Extracting periodicity of a regular texture based on autocorrelation functions. *Pattern Recognition Letters*, 18(5):433–443, 1997.
- [6] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Trans. on Systems, Man and Cybernetics*, 3(6):610–621, November 1973.
- [7] A.P Pentland. Fractal-based description of natural scenes. In *IEEE Trans. Patt. Anal. Mach. Intell.*, volume 6, pages 661–674, 1984.
- [8] H. Chen and W. Kinsner. Texture segmentation using multifractal measures. In *Proceedings of the Wescanex Conference on Communications, Power and Computing*, pages 222–227, 22-23 May 1997.
- [9] V. Anh, J. Maeda, M. Tieng, and H. Tsui. Multifractal texture analysis and classification. In *Proc. IEEE Int. Conf. Image Processing (ICIP)*, volume 4, pages 445–449, 1999.
- [10] M. Comer and E. Delp. Segmentation of textured images using a multiresolution gaussian autoregressive model. *IEEE Trans. Image Processing*, 8(3):408–420, March 1999.
- [11] G. Hazel. Multivariate gaussian MRF for multispectral scene segmentation and anomaly detection. *IEEE Trans. Geosci. Remote Sens.*, 38(3):1199–1211, May 2000.
- [12] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 1991.
- [13] T. P. Weldon, W. E. Higgins, and D. F. Dunn. Efficient Gabor filter design for texture segmentation. *Pattern Recognition*, 29(12):2005–2015, 1996.
- [14] P. Scheunders, S. Livens, G. Van De Wouwer, P. Vautrot, and D. Van Dyck. Wavelet-based texture analysis. *Int. Journal of Computer Science and Information Management, Special issue on Image Processing (IJCSIM)*, 1, 1998.

-
- [15] G. Van de Wouwer, P. Scheunders, and D. Van Dyck. Statistical texture characterization from discrete wavelet representation. *IEEE Trans. Image Processing*, 8(4):592–598, April 1999.
- [16] N. Sebe, M. S. Lew, and N. Bohrweg. Wavelet based texture classification. In *International Conference on Pattern Recognition*, pages 959–962, 2000.
- [17] A. Mojsilović, M. Popović, and D. Rachov. On the selection of an optimal wavelet basis for texture characterization. *IEEE Trans. Image Processing*, 9(12):2043–2050, December 2000.
- [18] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(7):971–987, 2002.
- [19] Y. Xia, D. D. Feng, and R. Zhao. Morphology-based multifractal estimation for texture segmentation. *IEEE Trans. Image Processing*, 15(3):614–623, 2006.
- [20] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, 1981.
- [21] D. Donoho. Compressive sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, April 2006.
- [22] N. E. Huang et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proc. of the Royal Society of London A: Math., Phys. and Engin. Sciences*, volume 454, pages 903–995, 1998.
- [23] P. Flandrin, G. Rilling, and P. Goncalves. Empirical mode decomposition as a filter bank. *IEEE Signal Processing Letters*, 11(2):112–114, 2004.
- [24] J. C. Nunes, Y. Bouaoune, E. Delechelle, O. Niang, and P. Bunel. Image analysis by bidimensional empirical mode decomposition. *Image and Vision Computing*, 21(12):1019–1026, 2003.
- [25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [26] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Computer vision—ECCV 2006*, pages 404–417. Springer, 2006.
- [27] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.
- [28] C. Shannon. Mathematical theory on communication. *Bell System Tech. J.*, 27:379, 623, 1948.
- [29] K. V. Mardia and P. E. Jupp. *Directional statistics*. John Wiley and Sons, Ltd, 2000.
- [30] J. Inglada and G. Mercier. A new statistical similarity measure for change detection in multitemporal SAR images and its extension to multiscale change analysis. *IEEE Trans. Geosci. Remote Sens.*, 45(5):1432–1445, 2007.
- [31] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Patt. Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

- [32] D. Huang, C. Zhu, Y. Wang, and L. Chen. HSOG: A novel local image descriptor based on histograms of the second order gradients. *IEEE Trans. Image Processing*, 23(11):4680–4695, 2014.
- [33] R. M. Haralick. Statistical and structural approaches to texture. *Proceeding. IEEE*, 67(5):786–804, 1979.
- [34] M. Pesaresi. Texture analysis for urban pattern recognition using fine-resolution panchromatic satellite imagery. *Geographical and Environmental Modelling*, 4(1):43–63, 2000.
- [35] F. Kayitakire, C. Hamel, and P. Defourny. Retrieving forest structure variables based on image texture analysis and ikonos-2 imagery. *Remote sensing of environment*, 102(3):390–401, 2006.
- [36] M. Hauta-Kasari, J. Parkkinen, T. Jaaskelainen, and R. Lenz. Multi-spectral texture segmentation based on the spectral cooccurrence matrix. *Pattern Analysis and Applications*, 2(4):275–284, 1999.
- [37] S. D. Newsam and C. Kamath. Retrieval using texture features in high-resolution multi-spectral satellite imagery. In *Defense and Security*, pages 21–32. International Society for Optics and Photonics, 2004.
- [38] L. K. Soh and C. Tsatsoulis. Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *Geoscience and Remote Sensing, IEEE Transactions on*, 37(2):780–795, 1999.
- [39] L. A. Ruiz, A. Fdez-Sarría, and J. A. Recio. Texture feature extraction for classification of remote sensing data using wavelet decomposition: a comparative study. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35:B4, 2004.
- [40] S. K. Meher, B. U. Shankar, and A. Ghosh. Wavelet-feature-based classifiers for multi-spectral remote-sensing images. *IEEE Trans. Geosci. Remote Sens.*, 45(6):1881–1886, 2007.
- [41] A. Lefebvre, T. Corpetti, and L. H. Moy. A measure for change detection in very high resolution remote sensing images based on texture analysis. In *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pages 1697–1700, 2009.
- [42] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
- [43] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *Signal Processing Magazine, IEEE*, 30(3):83–98, May 2013.
- [44] F. R. K. Chung. Spectral graph theory. In *AMS*, 1997.
- [45] U. V. Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
- [46] A. Ng, M. Jordan, and J. Malik. On the spectral clustering: Analysis and an algorithm. In *Proc. Adv. NIPS*, pages 849–856, 2001.

-
- [47] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C*, 28(1):100–108, 1979.
- [48] A. Sandryhaila and J. M. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013.
- [49] D. I. Shuman, B. Ricaud, and P. Vandergheynst. Vertex-frequency analysis on graphs. *Applied and Computational Harmonic Analysis*, 40(2):260–291, 2016.
- [50] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmon. Anal.*, 30(2):129–150, 2011.
- [51] A. Sanfeliu, R. Alquézar, J. Andrade, J. Climent, F. Serratosa, and J. Vergés. Graph-based representations and techniques for image processing and image analysis. *Pattern Recognition*, 35(3):639–650, 2002.
- [52] C. de Mauro, M. Diligenti, M. Gori, and M. Maggini. Similarity learning for graph-based image representations. *Pattern Recognition Letters*, 24(8):1115–1122, 2003.
- [53] P. C. Smits and A. Annoni. Updating land-cover maps by using texture information from very high-resolution space-borne imagery. *IEEE Trans. Geosci. Remote Sens.*, 37(3):1244–1254, 1999.
- [54] X. Wu, F. Yang, and R. Lishman. Land cover change detection using texture analysis. *Journal of Computer Science*, 6(1):92–100, 2010.
- [55] N. Li, H. Huo, and T. Fang. A novel texture-preceded segmentation algorithm for high-resolution imagery. *IEEE Trans. Geosci. Remote Sens.*, 48(7):2818–2828, 2010.
- [56] F. R. de Siqueira, W. R. Schwartz, and H. Pedrini. Multi-scale gray level co-occurrence matrices for texture description. *Neurocomputing*, 120:336–345, 2013.
- [57] Z. Hu, Z. Wu, Q. Zhang, Q. Fan, and J. Xu. A spatially-constrained color–texture model for hierarchical VHR image segmentation. *IEEE Geosci. Remote Sens. Lett.*, 10(1):120–124, 2013.
- [58] O. Rajadell, P. Garcia-Sevilla, and F. Pla. Spectral–spatial pixel characterization using gabor filters for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.*, 10(4):860–864, 2013.
- [59] J. Yuan, D. Wang, and R. Li. Remote sensing image segmentation by combining spectral and texture features. *IEEE Trans. Geosci. Remote Sens.*, 52(1), 2014.
- [60] S. Bernabe, P. R. Marpu, A. Plazao, M. D. Mura, and J. A. Benediktsson. Spectral-spatial classification of multispectral images using kernel feature space representation. *IEEE Geosci. Remote Sens. Lett.*, 11(1), 2014.
- [61] I. Jolliffe. *Principal component analysis*. Wiley Online Library, Ltd, 2002.
- [62] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM Journal on Matrix Analysis and Applications*, 13(1):357–385, 1992.
- [63] D. I. Shuman, P. Vandergheynst, and P. Frossard. Chebyshev polynomial approximation for distributed signal processing. In *Proc. IEEE Int. Conf. on Dist. Computing in Sensor Syst. and Wkshp. (DCOSS)*, pages 1–8, 2011.

- [64] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.
- [65] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.
- [66] G. Mercier, S. Derrode, E. Trouvé, and L. Bombrun. Change detection in remote sensing observations. *Multivariate Image Processing: Methods and Applications*, pages 95–142, 2010.
- [67] F. Bovolo and L. Bruzzone. A split-based approach to unsupervised change detection in large-size multitemporal images: Application to tsunami-damage assessment. *IEEE Trans. Geosci. Remote Sens.*, 45(6):1658–1670, 2007.
- [68] F. Del Frate, F. Pacifici, and D. Solimini. Monitoring urban land cover in Rome, Italy, and its changes by single-polarization multitemporal SAR images. *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, 1(2):87–97, 2008.
- [69] Y. Ban and O. Yousif. Multitemporal spaceborne SAR data for urban change detection in China. *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, 5(4):1087–1094, 2012.
- [70] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Trans. Image Processing*, 14(3):294–307, 2005.
- [71] E. J. M. Rignot and J. J. van Zyl. Change detection techniques for ERS-1 SAR data. *IEEE Trans. Geosci. Remote Sens.*, 31(4):896–906, 1993.
- [72] Y. Bazi, L. Bruzzone, and F. Melgani. An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images. *IEEE Trans. Geosci. Remote Sens.*, 43(4):874–887, 2005.
- [73] F. Bovolo and L. Bruzzone. A detail-preserving scale-driven approach to change detection in multitemporal SAR images. *IEEE Trans. Geosci. Remote Sens.*, 43(12):2963–2972, 2005.
- [74] J. Ma, M. Gong, and Z. Zhou. Wavelet fusion on ratio images for change detection in SAR images. *IEEE Geosci. Remote Sens. Lett.*, 9(6):1122–1126, 2012.
- [75] M. Gong, Z. Zhou, and J. Ma. Change detection in Synthetic Aperture Radar images based on image fusion and fuzzy clustering. *IEEE Trans. Image Processing*, 21(4):2141–2151, 2012.
- [76] B. Hou, Q. Wei, Y. Zheng, and S. Wang. Unsupervised change detection in SAR image based on Gauss-log ratio image fusion and compressed projection. *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, 7(8):3297–3317, 2014.
- [77] J. Inglada. Change detection on SAR images by using a parametric estimation of the Kullback-Leibler divergence. In *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 4104–4106. IEEE, 2003.
- [78] J. Zheng and H. You. A new model-independent method for change detection in multi-temporal SAR images based on Radon transform and Jeffrey divergence. *IEEE Geosci. Remote Sens. Lett.*, 10(1):91–95, 2013.

-
- [79] O. Yousif and Y. Ban. Improving urban change detection from multitemporal SAR images using PCA-NLM. *IEEE Trans. Geosci. Remote Sens.*, 51(4):2032–2041, 2013.
- [80] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, and F. Tupin. Change detection for high resolution satellite images, based on SIFT descriptors and an a contrario approach. In *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 1281–1284. IEEE, 2014.
- [81] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, and F. Tupin. SAR-SIFT: A SIFT-like algorithm for SAR images. *IEEE Trans. Geosci. Remote Sens.*, 53(1):453–466, 2015.
- [82] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [83] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern recognition*, 19(1):41–47, 1986.
- [84] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. *Computer Vision and Pattern Recognition*, 2(60), 2005.
- [85] O. Harant. *Méthodes statistiques en télédétection RSO polarimétrique haute résolution pour la modélisation et le suivi temporel des glaciers*. PhD thesis, Université Rennes 1, 2012.
- [86] R. Fallourd, O. Harant, E. Trouvé, J.M. Nicolas, M. Gay, A. Walpersdorf, J.L. Mugnier, J. Serafini, D. Rosu, and L. Bombrun. Monitoring temperate glacier displacement by multi-temporal TerraSAR-X images and continuous GPS measurements. *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, 4(2):372–386, 2011.
- [87] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV*, pages 589–600. Springer, 2006.
- [88] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng. Visual tracking via incremental log-euclidean riemannian subspace learning. In *Proc. IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, pages 1–8, 2008.
- [89] L. Dong, L. Tao, and G. Xu. Head pose estimation using covariance of oriented gradients. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, pages 1470–1473, 2010.
- [90] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR)*, volume 1, pages 886–893, 2005.
- [91] W. Förstner and B. Moonen. A metric for covariance matrices. In *Geodesy-The Challenge of the 3rd Millennium*, pages 299–309. Springer, 2003.
- [92] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao. WLD: A robust local image descriptor. *IEEE Trans. Patt. Anal. Mach. Intell.*, 32(9):1705–1720, 2010.
- [93] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
- [94] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. *Supervised machine learning: A review of classification techniques*. Informatica 31, 2007.

- [95] O. Regniers, L. Bombrun, D. Guyon, J. C. Samalens, and C. Germain. Wavelet-based texture features for the classification of age classes in a maritime pine forest. *IEEE Geosci. Remote Sens. Lett.*, 12(3):621–625, 2015.
- [96] O. Regniers, L. Bombrun, V. Lafon, and C. Germain. Supervised classification of very high resolution optical images using wavelet-based textural features. *IEEE Trans. Geosci. Remote Sens.*, 54(6):3722–3735, 2016.
- [97] T. A. Warner and K. Steinmaus. Spatial classification of orchards and vineyards with high spatial resolution panchromatic imagery. *Photogrammetric Engineering & Remote Sensing*, 71(2):179–187, 2005.
- [98] C. Delenne, G. Rabatel, and M. Deshayes. An automatized frequency analysis for vine plot detection and delineation in remote sensing. *IEEE Geosci. Remote Sens. Lett.*, 5(3):341–345, 2008.
- [99] G. Rabatel, C. Delenne, and M. Deshayes. A non-supervised approach using gabor filters for vine-plot detection in aerial images. *Computers and Electronics in Agriculture*, 62(2):159–168, 2008.
- [100] T. Ranchin, B. Naert, M. Albuissou, G. Boyer, and P. Astrand. An automatic method for vine detection in airborne imagery using the wavelet transform and multiresolution analysis. *Photogrammetric Engineering and Remote Sensing*, 67(1):91–98, 2001.
- [101] S. Cui, C. O. Dumitru, and M. Datcu. Ratio-detector-based feature extraction for very high resolution SAR image patch indexing. *IEEE Geosci. Remote Sens. Lett.*, 10(5):1175–1179, 2013.
- [102] O. Regniers, J. P. Da Costa, G. Grenier, C. Germain, and L. Bombrun. Texture based image retrieval and classification of very high resolution maritime pine forest images. In *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 4038–4041. IEEE, 2013.
- [103] T. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [104] S. Di Zenzo. A note on the gradient of a multi-image. *Computer vision, graphics, and image processing*, 33(1):116–125, 1986.
- [105] Vision texture. MIT Vision and Modeling group, Available online: <http://vismod.media.mit.edu/pub/VisTex/>.
- [106] R. Kwitt and P. Meerwald. Salzburg texture image database. Available online: <http://www.wavelab.at/sources/STex/>.
- [107] S. Abdelmounaime and H. Dong-Chen. New Brodatz-based image databases for grayscale color and multiband texture analysis. *ISRN Machine Vision*, 2013, 2013.

Evaluation criteria for detection tasks

This appendix provides the computation of several criteria used for the quantitative evaluation of a detection problem. Within the thesis, detection tasks have been appeared with two thematic applications: the land-cover change detection in Chapter 4 and the vine parcel detection in Chapter 7. In general, the result of a change detection problem can be considered as a binary classification result including two classes: *changed* and *unchanged*. Similarly, the result of vine detection task in Chapter 7 consists of two classes: *vine* and *non-vine*. Here, we provide the evaluation criteria for change detection problem. We note that these criteria are also used for vine detection task by simply considering the two classes *vine* and *non-vine* as classes *changed* and *unchanged*, respectively.

Let us remind some quantities resulted from a detection procedure:

- true positives (TP): the number of changed points correctly detected,
- true negatives (TN): the number of unchanged points correctly detected,
- false positives (FP): the number of unchanged points incorrectly detected as changed (i.e. false alarms),
- false negatives (FN): the number of changed points incorrectly detected as unchanged (i.e. missed detections).

Now, let N be the total number of keypoints considered for change detection algorithm, N_c and N_u be respectively the number of *changed* points and *unchanged* points from ground truth change map ($N_c + N_u = N$), the aforementioned indicators are calculated as follows:

- false alarm (FA) rate and good detection (GD) rate which are used to plot the ROCs:

$$P_{FA} = \frac{FP}{N_u}, \quad P_{GD} = \frac{TP}{N_c},$$

- percentage of total errors (TE) consisting of false alarms and missed detections:

$$P_{TE} = \frac{FP + FN}{N} \times 100\%,$$

- percentage of overall accuracy (OA):

$$P_{OA} = \frac{TP + TN}{N} \times 100\%.$$

Author Publications

Journal articles

- (J-6) M.T. Pham, G. Mercier, L. Bombrun, J. Michel, “Texture and color-based image retrieval using the local extrema features and Riemannian distance”, *IEEE Transaction on Image Processing*, in peer review.
- (J-5) M.T. Pham, G. Mercier, O. Regniers, J. Michel, “Texture retrieval from VHR optical remote sensed images using the local extrema descriptor with application to vineyard parcel detection”, *Remote Sensing*, vol. 8(5), 368, 2016.
- (J-4) M.T. Pham, G. Mercier, J. Michel, “PW-COG: an effective texture descriptor for VHR satellite imagery using a pointwise approach on covariance matrix of oriented gradients”, *IEEE Transaction on Geoscience and Remote Sensing (TGRS)*, vol. 54(6), pp. 3345-3359, 2016.
- (J-3) M.T. Pham, G. Mercier, J. Michel, “Change detection between SAR images using a pointwise approach and graph theory”, *IEEE Transaction on Geoscience and Remote Sensing (TGRS)*, vol. 54(4), pp. 2020-2032, 2016.
- (J-2) M.T. Pham, G. Mercier, J. Michel, “Pointwise graph-based local texture characterization for very high resolution multispectral image classification”, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)*, vol. 8(5), pp. 1962-1973, 2015.
- (J-1) M.T. Pham, G. Mercier, J. Michel, “Textural features from wavelets on graphs for very high resolution panchromatic Pléiades image classification”, *The French Journal of Photogrammetry and Remote Sensing (RFPT)*, n. 208, pp. 131-136, 2014.

Conference papers

- (C-5) M.T. Pham, G. Mercier, O. Regniers, L. Bombrun, J. Michel, “Texture retrieval from very high resolution remote sensing images using the local extrema-based descriptors”, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Beijing, China, 2016.
- (C-4) M.T. Pham, G. Mercier, J. Michel, “Covariance-based texture description from weighted coherence matrix and gradient tensors for polarimetric SAR image classification”, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Milan, Italy, 2015, pp. 2469-2472.

- (C-3) M.T. Pham, G. Mercier, J. Michel, “Pointwise approach on covariance matrix of oriented gradients for very high resolution image texture segmentation”, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Milan, Italy, 2015, pp. 1008-1011.
- (C-2) M.T. Pham, G. Mercier, J. Michel, “A keypoint approach for change detection between SAR images based on graph theory”, *8th IEEE International Workshop on the Analysis of Multitemporal Remote Sensing Images (Multi-Temp)*, Annecy, France, 2015, pp. 1-4.
- (C-1) M.T. Pham, G. Mercier, J. Michel, “Wavelets on graphs for very high resolution multispectral image texture segmentation”, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Quebec city, Canada, 2014, pp. 2273-2276.